

Tales Araujo Mendonça  
Bruno Gonçalves Araujo

# **Shell Linux**

## **Do Aprendiz ao Administrador**



editora  
**VIENA**

1ª Edição  
Bauru/SP  
Editora Viena  
2015



# Sumário

<b>Lista de Siglas e Abreviaturas.....</b>	<b>19</b>
<b>1. Introdução .....</b>	<b>21</b>
1.1. Hardware .....	23
1.1.1. Fonte, de Onde vem a Energia? .....	24
1.1.2. CPU (Central Processing Unit ou Unidade Central de Processamento) .....	24
1.1.3. Placa-mãe, Onboard e Offboard .....	24
1.1.4. Placa de Vídeo.....	24
1.1.5. Placa de Som.....	25
1.1.6. Placa de Rede.....	25
1.1.7. Memória ROM (Read-only Memory ou Memória Somente Leitura) .....	26
1.1.8. Memória Permanente .....	26
1.1.9. Memória RAM (Random Access Memory ou Memória de Acesso Aleatório).....	26
1.1.10. Gabinete e Cooler.....	27
1.2. Conhecendo o Ambiente Linux .....	28
1.3. Como Acessar o Shell .....	29
1.3.1. Principais Shells.....	29
1.3.2. Encontre o Shell .....	30
1.3.3. Entendendo a Interface do Shell .....	31
1.3.4. Shell e o Nosso Estudo.....	31
<b>2. Árvore de Diretórios .....</b>	<b>35</b>
2.1. O Que São Diretórios? .....	37
2.2. Entendendo a Estrutura de Diretórios .....	38
2.2.1. Diretório Raiz (/) .....	38
2.2.2. Diretório do Usuário (/home).....	38
2.2.3. Acesso a Outras Partições e Dispositivos (/media, /mnt) .....	38
2.2.4. Organização de Arquivos dos Programas.....	38
2.2.4.1. Entenda a Separação dos Diretórios Executáveis .....	39
2.2.4.2. Executáveis do Sistema (/bin, /usr/bin, /usr/sbin/, /usr/local/bin, /sbin /usr/games) .....	39
2.2.5. Configurações do Sistema (/etc).....	40
2.2.6. Recursos do Sistema Unix (/usr) .....	40
2.2.7. Biblioteca dos Aplicativos (/lib, /usr/lib, /usr/local/lib) .....	40
2.2.8. Diretório Temporário (/tmp) .....	40
2.2.9. Variáveis do Sistema e Arquivos de Log (/var) .....	40
2.2.10. Diretórios Especiais (/dev, /proc, /sys).....	41
2.2.11. Diretório de Boot (/boot) .....	41
2.2.12. Diretório de Armazenamento de Programas Opcionais (/opt).....	41
<b>3. Principais Comandos .....</b>	<b>45</b>
3.1. Atalhos do Teclado.....	47
3.1.1. Navegando com a Tecla <Tab> .....	47
3.1.2. Histórico do Terminal, Setas Direcionais e Pesquisa .....	50

3.1.3.	Limpar o Histórico do Shell.....	50
3.1.4.	Interrompendo ou Congelando um Comando em Execução.....	50
3.2.	Gerenciamento de Arquivos e Diretórios (ls, cd, pwd, mkdir, rm, cp, mv) .....	51
3.2.1.	Listar Arquivo (ls) .....	51
3.2.2.	Acessar Diretório (cd) .....	53
3.2.3.	Diretório Corrente (pwd) .....	54
3.2.4.	Criar Diretório (mkdir) .....	55
3.2.5.	Remover Arquivo (rm) .....	56
3.2.6.	Copiar Arquivo (cp) .....	57
3.2.7.	Mover ou Renomear Arquivo (mv) .....	58
3.3.	Obtendo Informações para Usuários (date, head, tail, more, less, who) .....	59
3.3.1.	Exibir ou Configurar Data e Hora (date) .....	60
3.3.2.	Exibir as Primeiras Linhas de um Arquivo (head) .....	62
3.3.3.	Exibir as Últimas Linhas de um Arquivo (tail) .....	63
3.3.4.	Visualizador de Texto (more) .....	63
3.3.5.	Visualizador de Texto (less) .....	67
3.3.6.	Mostrar quem está Logado (who) .....	69
3.4.	Ler Informações do Hardware (/proc) .....	70
3.5.	Procurar Arquivos (find, locate) .....	72
3.5.1.	Procura por Arquivo (find) .....	72
3.5.1.1.	Dicas Sobre o Find .....	73
3.5.2.	Procurar por Arquivo em Banco de Dados (locate) .....	74
3.6.	Desligar e Reiniciar a Máquina (shutdown e reboot) .....	75
3.6.1.	Desligar ou Reiniciar o Sistema (shutdown) .....	75
3.6.2.	Desligar ou Reiniciar o Sistema (reboot) .....	77
3.7.	Modificar Parâmetros de Arquivos, como Data de Acesso (touch) .....	77
3.8.	Montar e Desmontar Partições (mount / umount) .....	79
3.9.	Juntar, Compactar e Descompactar Arquivos (tar, gunzip, bzip2) .....	82
3.9.1.	Juntar Arquivos com tar .....	82
3.9.2.	Compactar Arquivos com gunzip e tar .....	84
3.9.3.	Compactar Arquivos com bzip2 e tar .....	85
3.10.	Operações Matemáticas .....	85
3.10.1.	Operações Matemáticas Utilizando bc .....	86
<b>4.</b>	<b>Atributos Introdutórios ao Shell .....</b>	<b>93</b>
4.1.	Processos .....	95
4.1.1.	Visualizar os Processos .....	95
4.1.2.	Matar Processos .....	96
4.2.	Redirecionamentos (stdout, stderr, >, 2>, >>, 2>>, &>, &>>) .....	97
4.2.1.	Redirecionando com o pipe ( ) .....	99
4.3.	Variáveis .....	100
4.4.	Variáveis de Ambiente .....	101
4.5.	Usuários .....	103
4.6.	Grupos .....	104
4.6.1.	Criar e Remover um Grupo .....	105
4.7.	Permissões (chown, chmod, root e sudo) .....	106
4.7.1.	Permissões de Usuários e Grupos (chown) .....	106

4.7.2.	Permissões de Arquivos (chmod) .....	107
4.7.3.	Comando sudo e Usuário root.....	109
4.8.	Usando Aspas, Apóstrofes e Barra Invertida.....	110
4.9.	Ponto e Vírgula .....	112
4.10.	O Uso de Crase e Parênteses .....	112
4.11.	Delimitadores .....	113
4.12.	Principais Caracteres Especiais (Expressões Regulares).....	113
<b>5.</b>	<b>Aplicativos para o Dia a Dia.....</b>	<b>121</b>
5.1.	Fazer Download de Arquivos e Páginas (wget) .....	123
5.2.	Acessar Chats pelo Terminal (CenterIM).....	125
5.3.	Ouvir Músicas pelo Terminal (mpg123).....	127
5.4.	Assistir a Vídeos em ASCII pelo Terminal (MPlayer).....	128
5.5.	Editar Arquivos (vim, nano).....	129
5.5.1.	Editor de Textos vim .....	129
5.5.2.	Editor de Textos nano .....	131
5.6.	Checar a Integridade de Arquivos (md5sum) .....	133
5.7.	Ajustar o Volume (alsamixer).....	134
5.8.	Baixar Vídeos do YouTube (youtube-dl) .....	135
5.9.	Acessar o Servidor via ssh.....	140
5.9.1.	Copiar Arquivos via ssh .....	141
5.10.	Navegar na Internet pelo Modo Texto (links2).....	142
5.11.	Converter e Redimensionar Imagens (convert).....	143
<b>6.</b>	<b>Monitorando o Sistema .....</b>	<b>147</b>
6.1.	Processos em Execução (top).....	149
6.1.1.	Tempo de Atividade Média e Carga do Sistema .....	149
6.1.2.	Tarefas em Execução .....	150
6.1.3.	Estados da CPU.....	150
6.1.4.	Uso da Memória.....	150
6.1.5.	Identificação em Colunas .....	151
6.1.6.	Comandos Interativos.....	151
6.2.	Entenda o Uso da Memória (free -m) .....	152
6.3.	Uso de Disco (iotop, du, df) .....	153
6.3.1.	Análise de Escrita e Leitura do Disco (iotop).....	153
6.3.2.	Estimar Espaço Ocupado por Arquivo (du).....	154
6.3.3.	Estimar Espaço Ocupado por Partições (df) .....	155
6.4.	Informações Sobre a Conexão (ifconfig, iwlist).....	156
6.4.1.	Configurar Rede (ifconfig) .....	157
6.4.2.	Informação Sobre Rede Wireless (iwlist) .....	158
6.5.	Veja os Dispositivos da Máquina (lspci, lsusb) .....	161
6.5.1.	Informação Sobre Dispositivos (lspci).....	161
6.5.2.	Informações Sobre Dispositivos USB (lsusb) .....	166
6.6.	Monitore a Temperatura (sensors e hddtemp) .....	167
6.6.1.	Monitore a Temperatura com o sensors.....	167
6.6.2.	Monitore a Temperatura do HD com o hddtemp .....	168
6.7.	Informações Gerais (inxi) .....	168

<b>7.</b>	<b>Manipulação de Pacotes .....</b>	<b>173</b>
7.1.	O que é um Pacote .....	175
7.2.	O que são Repositórios .....	175
7.3.	O que são Gerenciadores de Pacotes .....	175
7.4.	Debian, Ubuntu e Derivados (DPKG, Apt-Get) .....	176
7.4.1.	Atualizar Repositório .....	177
7.4.2.	Atualizar Sistema .....	177
7.4.3.	Pesquisar por Pacotes (APT) .....	177
7.4.4.	Instalar Pacote (APT) .....	178
7.4.5.	Remover Pacote (APT) .....	179
7.4.6.	Remover Pacotes não Utilizados .....	179
7.4.7.	Limpar Cache de Pacotes Baixados .....	180
7.4.8.	Dicas Gerais de APT e DPKG .....	180
7.4.8.1.	Exibir todos os Pacotes Instalados no Sistema .....	180
7.4.8.2.	Verificar se um Pacote está Instalado .....	180
7.4.8.3.	Exibir a Lista de Arquivos Contidos em um Pacote Instalado .....	180
7.4.8.4.	Pesquisar a qual Pacote Pertence um Arquivo .....	180
7.4.8.5.	Instalar Pacote (DPKG) .....	180
7.4.8.6.	Instalar Pacote ".deb" local Satisfazendo as Dependências .....	181
7.4.8.7.	Remover Pacote (DPKG) .....	181
7.4.8.8.	Baixar Código-Fonte do Aplicativo .....	181
7.4.8.9.	Reconfigurar um pacote instalado .....	181
7.4.8.10.	Concluir a Configuração de Pacotes Interrompidos Durante a Instalação .....	181
7.4.8.11.	Reparar o Sistema com Pacotes Quebrados .....	181
7.4.8.12.	Repositórios .....	182
7.5.	RedHat, Fedora, CentOS (RPM, YUM) .....	182
7.5.1.	Verificar se o Pacote está Instalado (RPM) .....	183
7.5.2.	Instalar um Pacote (RPM) .....	183
7.5.3.	Obter Informações de um Pacote (RPM) .....	183
7.5.4.	Remover um Pacote (RPM) .....	184
7.5.5.	Lidar com Dependências de um Pacote (RPM) .....	185
7.5.6.	Instalar um Pacote e Lidar com Dependências (YUM) .....	185
7.5.7.	Remover um Pacote (YUM) .....	188
7.5.8.	Pesquisar um Pacote (YUM) .....	189
7.5.9.	Atualizar todo o Sistema (YUM) .....	190
7.5.10.	Instalar Grupo de Pacotes (YUM) .....	190
7.6.	Sabayon (Entropy) .....	190
7.6.1.	Atualizar Repositórios .....	191
7.6.2.	Atualizar o Sistema .....	191
7.6.3.	Pesquisar Pacote .....	191
7.6.4.	Instalar Pacote .....	192
7.6.5.	Remover Pacote .....	193
7.6.6.	Limpar Cache de Pacotes Recebidos .....	193
7.6.7.	Atualizar Configurações de Pacotes que Foram Instalados .....	193
7.6.8.	Checar por Dependências em Falta e Concertar .....	194

7.6.9.	Checar por Bibliotecas em Falta e Concertar.....	194
7.6.10.	Dicas de Atalhos.....	194
<b>8.</b>	<b>Manipular Textos de Forma não Interativa .....</b>	<b>199</b>
8.1.	Exibir Textos na Tela (echo, cat, tac, rev) .....	201
8.1.1.	Exibir Texto na Tela (echo) .....	201
8.1.2.	Exibir Conteúdo de um Arquivo ou Concatenar Texto (cat) .....	201
8.1.3.	Exibir as Linhas do Texto de Forma Invertida (tac) .....	203
8.1.4.	Inverter os Caracteres de cada Linha (rev) .....	203
8.2.	Filtrar Texto (grep e egrep) e Exemplos de Expressões Regulares .....	204
8.2.1.	Filtrar uma Opção ou Outra, o uso do Pipe ( ) .....	204
8.2.2.	Filtrar Qualquer Caractere e Escapando dos Meta-Caracteres, o Uso do Ponto e da Barra Invertida (\) .....	205
8.2.3.	Repetir ao Infinito e Além, o Uso do Asterisco (*) .....	206
8.2.4.	Repetir com o Uso do Caractere Soma (+) .....	207
8.2.5.	Caractere Opcional, a Interrogação (?) .....	208
8.2.6.	Fazer uma Lista, o Uso dos Colchetes .....	208
8.2.7.	Fazer uma Lista de Negação, os Colchetes com Circunflexo .....	209
8.2.8.	Filtrar o Início da Linha com o Circunflexo (^) .....	209
8.2.9.	Filtrar o Fim da Linha com o Cifrão (\$) .....	210
8.2.10.	O Uso da Borda \b .....	210
8.2.11.	O Uso da Contra Borda \B .....	210
8.2.12.	Estipular a Quantidade de Repetições com as Chaves .....	210
8.3.	Os Parâmetros do Grep e Egrep .....	211
8.3.1.	Contar Linhas, o Parâmetro -c .....	211
8.3.2.	Exibir o Número da Linha no Resultado, o Parâmetro -n .....	211
8.3.3.	Inverter o Resultado, Exibindo Apenas as Linhas que não Coincidem com o que Procuramos, o Parâmetro -v .....	211
8.3.4.	Exibir Algumas Linhas Após ou Antes dos Resultados, os Parâmetros -A e -B .....	212
8.3.5.	Ignorar Diferença entre Maiúscula e Minúscula, o Parâmetro -i .....	212
8.3.6.	Utilizar o Grep em Conjunto com Outros Comandos, o Uso do Pipe ( ) .....	212
8.4.	Recortar Pedacos de Textos com o Comando Cut .....	213
8.5.	Editar Textos com o Sed .....	215
8.5.1.	Apagar Linhas e Caracteres com o Sed .....	216
8.5.2.	Substituir Caracteres com o Sed .....	218
8.5.3.	Mais Alguns Exemplos do Sed .....	222
8.5.4.	Dicas Complementares Sobre Sed .....	224
8.6.	Editar Textos com o awk .....	226
8.6.1.	Alterar os Delimitadores do awk .....	227
8.6.2.	Filtrar Valores com o awk .....	228
8.7.	Exibir Textos em Ordem Alfabética e Remover Linhas Duplicadas (sort) .....	229
8.8.	Conclusão Sobre a Manipulação de Texto de Forma Não Interativa .....	230
<b>9.</b>	<b>Shell Script .....</b>	<b>235</b>
9.1.	Criar o Primeiro Shell Script e Entender os Comentários .....	238
9.2.	Passar Parâmetros para o Shell Script .....	240

<b>10.</b>	<b>Comandos Condicionais .....</b>	<b>245</b>
10.1.	If, Then, Else .....	247
10.1.1.	Verificar Strings .....	247
10.1.1.1.	Verificar se uma Variável não é Nula, -n.....	247
10.1.1.2.	Verificar se uma Variável é Nula, -z.....	248
10.1.1.3.	Conferir se é Igual.....	249
10.1.1.4.	Conferir se é Diferente.....	249
10.1.2.	If, Then, Else, o que Fazer Quando a Condição não for Verdadeira.....	250
10.1.2.1.	If, Then, Else e as Variáveis .....	250
10.1.3.	Verificações Numéricas .....	251
10.1.3.1.	Igual ou Diferente, Exclusivo para Números, -eq e -ne.....	251
10.1.3.2.	Menor que -lt .....	252
10.1.3.3.	Maior que -gt .....	253
10.1.3.4.	Menor ou Igual -le .....	253
10.1.3.5.	Maior ou Igual -ge .....	254
10.1.4.	Teste de Arquivos .....	255
10.1.4.1.	É um Diretório -d .....	255
10.1.4.2.	É um Arquivo -f.....	255
10.1.4.3.	Existência de Diretório ou Arquivo -e .....	255
10.1.4.4.	O Arquivo ou Diretório Pertence a um Grupo a que o Usuário Também Pertence -G.....	256
10.1.4.5.	O Arquivo ou Diretório Pertence ao Usuário -O.....	256
10.1.4.6.	O Arquivo ou Diretório Possui Permissão de Leitura -r.....	256
10.1.4.7.	O Arquivo ou Diretório Possui Permissão de Escrita -w.....	256
10.1.4.8.	O Arquivo ou Diretório Possui Permissão de Execução -x.....	256
10.1.4.9.	O Arquivo foi Criado ou Modificado Depois -nt.....	257
10.1.4.10.	O Arquivo foi Criado ou Modificado Antes -nt.....	257
10.1.5.	Invertendo a Checagem com o Uso da Exclamação (!).....	257
10.1.6.	Checagens Múltiplas && e    .....	258
10.1.6.1.	Exigir que Todas as Ocorrências Sejam Verdadeiras && (e) .....	258
10.1.6.2.	Permitir que Apenas uma das Ocorrências seja Verdadeira    (ou).....	259
10.1.6.3.	Checagens && e    Podem Trabalhar em Conjunto.....	259
10.1.7.	Outra Forma de Múltiplas Checagens, elif .....	260
10.2.	O Uso do Loop, ou Laço, Resolvendo Problemas Repetitivos .....	260
10.2.1.	Utilizar o Laço for.....	261
10.2.1.1.	Utilizar o Laço for com Comandos no Lugar de Listas .....	262
10.2.2.	Utilizar o Laço while.....	264
10.2.2.1.	Utilizar o Laço While para Fazer um Loop Infinito .....	266
10.2.3.	Utilizar o Laço until .....	266
10.3.	Função .....	267
10.4.	Casando com o Case .....	267
<b>11.</b>	<b>Interação com Shell Script .....</b>	<b>275</b>
11.1.	Read.....	277
11.2.	Dicas Gerais para Diálogos.....	278
11.2.1.	Descobririndo se o Usuário Marcou Sim/Ok ou Não/ Cancelar.....	278
11.3.	Dialog.....	279



11.3.1.	Mensagem na Tela (--msgbox).....	279
11.3.2.	Mensagem sem Botões (--infobox).....	280
11.3.3.	Ler um Arquivo (--textbox).....	281
11.3.4.	Monitoramento com tail -f (--tailbox).....	281
11.3.5.	Entre com os Dados (--inputbox).....	282
11.3.6.	Pergunta Sim ou Não (--yesno).....	283
11.3.7.	Menu de Opções (--menu).....	283
11.3.8.	Escolha uma Opção (--radiolist).....	284
11.3.9.	Calendário (--calendar).....	285
11.3.10.	Lista de Opções (--checklist).....	285
11.3.11.	Selecione um Arquivo (--fselect).....	286
11.3.12.	Escolha um Horário (--timebox).....	287
11.3.13.	Exemplo de Script Utilizando Dialog.....	287
11.4.	Zenity.....	289
11.4.1.	Diálogo de Informação (--info).....	289
11.4.2.	Diálogo de Aviso (--warning).....	290
11.4.3.	Diálogo de Erro (--error).....	291
11.4.4.	Diálogo de Notificação (--notification).....	291
11.4.5.	Diálogo de Pergunta (--question).....	292
11.4.6.	Diálogo de Entrada de Texto (--entry).....	292
11.4.7.	Diálogo Usuário e Senha (--password).....	294
11.4.8.	Diálogo de Formulário (--forms).....	294
11.4.9.	Diálogo de Texto Informativo (--text-info).....	296
11.4.10.	Diálogo de Lista (--list).....	297
11.4.11.	Diálogo de Calendário (--calendar).....	299
11.4.12.	Diálogo de Seleção de Arquivo (--file-selection).....	299
11.4.13.	Diálogo de Escala (--scale).....	301
11.4.14.	Diálogo de Seleção de Cor (--color-selection).....	301
11.4.15.	Exemplo de Script Utilizando Zenity.....	302
11.5.	Kdialog.....	305
11.5.1.	Caixa de Mensagem (--msgbox).....	305
11.5.1.1.	Caixa de Mensagem de Desculpas (--sorry).....	305
11.5.1.2.	Caixa de Mensagem de Erro (--error).....	306
11.5.2.	Pergunta Sim ou Não (--yesno).....	306
11.5.3.	Diálogos de Entrada do Usuário (--inputbox).....	307
11.5.3.1.	Diálogos de Entrada do Usuário (--password).....	308
11.5.4.	Exibir Arquivo Texto (--textbox).....	308
11.5.5.	Menu Opções (--menu).....	309
11.5.6.	Lista de Opções (--checklist).....	310
11.5.7.	Escolha uma Opção (--radiolist).....	311
11.5.8.	Selecionar Arquivo (--getopenfilename).....	311
11.5.8.1.	Salvar Arquivo (--getsavefilename).....	313
11.5.8.2.	Selecionar Pasta (--getexistingdirectory).....	314
11.5.9.	Controle de Ajuste (--slider).....	315
11.5.10.	Diálogo de Seleção de Cor (--getcolor).....	316

11.5.11.	Exemplo de Script Utilizando Kdialog .....	316
11.6.	Outras Interfaces para o Shell, Xdialog, Yab e BigBashView .....	317
<b>12.</b>	<b>Executando Shell Script Automaticamente no Boot .....</b>	<b>321</b>
12.1.	SysVinit .....	323
12.2.	Upstart .....	323
12.3.	Systemd .....	324
<b>13.</b>	<b>Recursos de Shell Script na Interface Gráfica.....</b>	<b>329</b>
13.1.	Gerenciadores de Arquivos (Dolphin e Nautilus).....	331
13.2.	Editores de Textos (Kate e Gedit) .....	331
<b>14.</b>	<b>Shell e Dispositivos Móveis .....</b>	<b>333</b>
14.1.	Utilizando Shell no seu Smartphone ou Tablet Android.....	335
14.2.	Linux em Outros Sistemas para Smartphone.....	337
<b>15.</b>	<b>Mais Alguns Conceitos .....</b>	<b>339</b>
15.1.	Características de um Sistema Operacional Linux.....	341
15.2.	Licença GPL.....	341
15.3.	Projeto GNU.....	342
15.4.	Software Livre .....	342
15.5.	GNU/LINUX, Projetos Complementares .....	343
15.6.	Padrões Abertos e Interoperabilidade .....	344
15.7.	Software Legado.....	345
<b>16.</b>	<b>Chegamos ao Fim .....</b>	<b>349</b>
<b>Referências.....</b>		<b>363</b>
<b>Glossário .....</b>		<b>365</b>

# Lista de Siglas e Abreviaturas

<i>ABNT</i>	<i>Associação Brasileira de Normas Técnicas.</i>
<i>APT</i>	<i>Advanced Packaging Tool.</i>
<i>ASCII</i>	<i>American Standard Code for Information Interchange.</i>
<i>BIOS</i>	<i>Basic Input/Output System.</i>
<i>BSD</i>	<i>Berkeley Software Distribution.</i>
<i>CPU</i>	<i>Central Processing Unit.</i>
<i>CSS</i>	<i>Cascading Style Sheets.</i>
<i>DDR</i>	<i>Double Data Rate.</i>
<i>DVD</i>	<i>Digital Versatile Disc.</i>
<i>EFI</i>	<i>Extensible Firmware Interface.</i>
<i>FSF</i>	<i>Free Software Foundation.</i>
<i>FTP</i>	<i>File Transfer Protocol.</i>
<i>GB</i>	<i>Gigabyte.</i>
<i>GIF</i>	<i>Graphics Interchange Format.</i>
<i>GPL</i>	<i>General Public License.</i>
<i>GPU</i>	<i>Graphics Processing Unit.</i>
<i>GRUB</i>	<i>Grand Unified Bootloader.</i>
<i>GUI</i>	<i>Graphical User Interface.</i>
<i>HD</i>	<i>Hard Disk.</i>
<i>HTML</i>	<i>HyperText Markup Language.</i>
<i>I/O</i>	<i>Input/Output.</i>
<i>ISO</i>	<i>International Organization for Standardization.</i>
<i>JPEG</i>	<i>Joint Photographics Experts Group.</i>
<i>KBPS</i>	<i>Quilobit Por Segundo.</i>
<i>Mac OS</i>	<i>Macintosh Operating System.</i>
<i>MB</i>	<i>Megabyte.</i>
<i>MKSH</i>	<i>MirBSD Korn Shell.</i>
<i>MP3</i>	<i>Moving Picture Experts Group Audio Layer 3.</i>
<i>MP4</i>	<i>Moving Picture Experts Group Audio Layer 4.</i>
<i>MTU</i>	<i>Maximum Transmission Units.</i>
<i>ODF</i>	<i>Open Document Format for Office Applications.</i>
<i>OpenCL</i>	<i>Open Computing Language.</i>
<i>PCI</i>	<i>Peripheral Component Interconnect.</i>
<i>PID</i>	<i>Process ID.</i>
<i>PNG</i>	<i>Portable Network Graphics.</i>
<i>PPM</i>	<i>Parte Por Milhão.</i>
<i>PTS</i>	<i>Pseudo-Terminal Slave.</i>
<i>RAM</i>	<i>Random Access Memory.</i>
<i>ROM</i>	<i>Read-only Memory.</i>
<i>RPM</i>	<i>RPM Package Manager.</i>
<i>RWX</i>	<i>Read, Write, Execute.</i>
<i>SH</i>	<i>Shell Script.</i>
<i>SSD</i>	<i>Solid-State Drive.</i>

<i>SSH</i>	<i>Secure Shell.</i>
<i>TTY</i>	<i>Teletypewriter.</i>
<i>UEFI</i>	<i>Unified Extensible Firmware Interface.</i>
<i>URL</i>	<i>Universal Resource Locator.</i>
<i>USB</i>	<i>Universal Serial Bus.</i>
<i>USR</i>	<i>Unix System Resources.</i>
<i>WWW</i>	<i>World Wide Web.</i>
<i>XMPP</i>	<i>Extensible Messaging and Presence.</i>
<i>YUM</i>	<i>Yellow dog Updater Modified.</i>

# 1

## Introdução

### 1.1. Hardware

- 1.1.1. Fonte, de Onde vem a Energia?
- 1.1.2. CPU (Central Processing Unit ou Unidade Central de Processamento)
- 1.1.3. Placa-Mãe, Onboard e Offboard
- 1.1.4. Placa de Vídeo
- 1.1.5. Placa de Som
- 1.1.6. Placa de Rede
- 1.1.7. Memória ROM (Read-only Memory ou Memória Somente Leitura)
- 1.1.8. Memória Permanente
- 1.1.9. Memória RAM (Random Access Memory ou Memória de Acesso Aleatório)
- 1.1.10. Gabinete e Cooler

### 1.2. Conhecendo o Ambiente Linux

### 1.3. Como Acessar o Shell

- 1.3.1. Principais Shells
- 1.3.2. Encontre o Shell
- 1.3.3. Entendendo a Interface do Shell
- 1.3.4. Shell e o Nosso Estudo



# 1. Introdução

Provavelmente você já viu, pelo menos em filmes, alguém utilizando um computador onde a interface é uma tela preta em que a pessoa digita comandos e recebe respostas. Hoje isso parece algo completamente obsoleto, mas tudo que seu computador faz ainda é responder a comandos, porém você não os digita, mas utiliza ícones e outros tipos de interfaces para facilitar a interação com o computador. Por trás de todo o funcionamento, o computador está executando comandos e enviando respostas.

Este livro irá ensiná-lo a utilizar o computador de forma mais direta, ultrapassar a camada da interface e comandá-lo diretamente por meio de um terminal de comandos **Linux**, chamado de **Shell**.

Antes de prosseguir é importante responder a algo que provavelmente você esteja se perguntando! Se existem opções mais simples de interagir com o computador, por que utilizar comandos?

Compreender o uso do **Shell** vai ajudar você a compreender como os computadores funcionam e como ser mais produtivo, principalmente para tarefas repetitivas. Este livro irá mostrar como criar formas de automação que simplificam nosso dia a dia. Todos que são apaixonados por informática devem aprender sobre **Shell**, mas se você trabalha com informática, gerenciando um parque de computadores, servidores, redes ou uma série de outras funções, saber interagir da melhor forma com o **Shell** pode ser o grande diferencial que fará você ser produtivo como nunca.

Antes de entrarmos realmente no mundo **Shell Linux**, é interessante conhecer um pouco sobre como a parte física do computador funciona, conhecer um pouco sobre o hardware, as peças que compõem o computador, alguns componentes principais e como esse conjunto de peças torna possível a interação do homem com a máquina.

Quando lembramos de computador, tendemos a lembrar do aparelho que possui gabinete e monitor ou então de notebooks, porém grande parte dos produtos hoje são computadores ou possuem um computador embutido: smartphones, roteadores, videogames, leitores de blu-ray, TVs inteligentes e uma série de outros dispositivos possuem um computador embutido. Em muitos casos é somente um computador com outra aparência. Muitos desses computadores utilizam Linux, e podemos até acessar o terminal **Shell** deles, como é o caso dos smartphones com Android.

## 1.1. Hardware

O **hardware** é toda a parte física do computador, ou seja, são as peças que unidas fazem o computador funcionar. Este não é um livro destinado a ensinar sobre o hardware, portanto essa parte será apresentada de forma bastante simplificada para que seja compreendida apenas a lógica do funcionamento. Para isso faremos um resumo do funcionamento de um computador simples, com as seguintes partes: Fonte, CPU, placa-mãe, placa de vídeo, placa de som, placa de rede, memória permanente, memória volátil (RAM), processador, leitor de disco, gabinete, monitor e a diferença de onboard e offboard.

### **1.1.1. Fonte, de Onde vem a Energia?**

Os componentes do computador dependem de energia para o seu funcionamento. A fonte transforma a energia da tomada, que é de corrente alternada, em corrente contínua e também assegura que a corrente e a voltagem serão adequadas para ligar cada um dos componentes que farão o computador funcionar.

### **1.1.2. CPU (Central Processing Unit ou Unidade Central de Processamento)**

O processador é onde os dados são manipulados, que realiza os cálculos e toma as decisões. Tudo que fazemos em um computador passa pelo processador, é como se ele fosse o cérebro do computador. Os processadores estão cada vez mais complexos e até para definir suas capacidades, atualmente, é bastante difícil; houve um tempo em que sua frequência era utilizada como base para saber a velocidade, medida realizada geralmente em Megahertz (MHz) ou Gigahertz (GHz). Atualmente também devemos observar quantos núcleos possui, qual arquitetura e uma série de outras informações.

### **1.1.3. Placa-mãe, Onboard e Offboard**

A placa-mãe é a maior placa, o maior componente interno que compõe o computador. É nela que todas as outras peças são encaixadas, como: fonte de energia, CPU, memórias, placa de vídeo, placa de som, placa de rede, disco de dados (HD ou SSD). Existem alguns componentes citados que podem vir acoplados na placa-mãe, é o que chamamos de placas onboard, ou seja, placas que estão inclusas em outra. Quando precisamos encaixar outra placa, chamamos de placa offboard, também chamada de placa dedicada, pois ela se dedica a efetuar apenas uma função.

A função básica da placa-mãe é interligar todos os outros componentes, porém com o passar do tempo diversos dispositivos foram incluídos nessa placa, o que é útil tanto para reduzir custos quanto para economizar espaço. Atualmente quase todas as placas-mãe possuem placa de som e placa de rede inclusas. Na placa-mãe também encontramos os espaços para encaixar o processador, memória RAM, HD, conectores USB e diversas outras opções que variam de acordo com o modelo da placa.

Em geral os dispositivos onboard atuais possuem boa qualidade, mas, quando se trata de placa de vídeo, para obter bom desempenho para trabalhos 3D ou jogos de alta performance, é preciso instalar uma placa de vídeo à parte, ou seja, uma placa de vídeo offboard.

### **1.1.4. Placa de Vídeo**

Ao ligar o monitor do seu computador ou ligar seu notebook e visualizar uma imagem na tela, é a placa de vídeo que faz essa “mágica” acontecer. Ela é responsável por transformar os pulsos elétricos que são transmitidos como instrução da CPU para a placa de vídeo, para o monitor de vídeo, possibilitando que imagens sejam geradas em tela.

As placas de vídeo são essenciais para que se obtenha qualidade gráfica de ponta, como os jogos e animações 3D, elas são praticamente computadores dentro do computador, possuem seus próprios processadores, que são as unidades de processamento gráfico



ou GPU. Quando são offboard também possuem memória **RAM** própria, que tende a ser de velocidade bem mais rápida que a memória normal do sistema, e geralmente possuem até sistema próprio de resfriamento. Em placas de vídeo modernas também é bastante comum que elas possuam uma placa de som inclusa.

A capacidade das placas de vídeo cresceu tanto que atualmente existem projetos com o **OpenCL**, que se dedicam a aperfeiçoar técnicas de criação de programas que utilizem também a capacidade da placa de vídeo para outras atividades, como a geração de chaves criptográficas.

### 1.1.5. Placa de Som

Assim como a placa de vídeo possibilita que imagens sejam geradas em uma tela, a placa de som possibilita que sons sejam emitidos por alto-falantes. O funcionamento da maioria das placas que são inseridas na placa-mãe é semelhante no quesito processamento de fluxo. Há um envio de pulsos elétricos que são decodificados pela **CPU** e transmitidos para a placa de som, possibilitando a emissão do ruído sonoro.

As placas de som realmente dedicadas se tornaram um tanto raras, algo quase exclusivo para quem trabalha com áudio, pois, como já foi dito, quase toda placa-mãe possui placa de som embutida, algumas placas de vídeo também, e até mesmo alguns fones de ouvido **USB** já possuem suas próprias placas de som inclusas. Ainda existem os fones de ouvido bluetooth, que funcionam como se fosse uma placa de som a mais no computador.

### 1.1.6. Placa de Rede

Placas de rede servem para que os computadores possam se comunicar entre si. O funcionamento das placas de rede se dá da mesma maneira como a placa de som e a placa de vídeo, por meio de pulsos elétricos passados pela **CPU**. É essa placa que possibilita a você acessar a internet ou algum outro computador em rede. Podemos dividir as placas de rede em dois tipos, cabeadas e wireless (sem fio).

Hoje em dia a maioria das pessoas não “sobrevivem” sem uma placa de rede, pois não ter uma é estar desconectado da rede mundial de computadores, a **WWW (World Wide Web)**, o que para muitos é uma grande aflição. Hoje, tudo gira em torno da internet, trabalhos e quase tudo pode ser resolvido sem sair de casa, usando apenas a internet.

Em computadores domésticos geralmente existe apenas uma placa de rede para uso com cabo e em alguns casos também uma placa para uso sem fios, mas em servidores é normal que se tenha diversas placas de rede.

Notebooks e smartphones praticamente todos possuem placas de rede wireless, porém, quando se fala em placa wireless, costuma ser uma referência somente aos dispositivos compatíveis com o padrão 802.11, que atualmente se divide principalmente nas categorias b, g, n. Esses padrões e siglas servem para verificar se existe compatibilidade de comunicação entre os equipamentos.

Também podemos considerar como placas de rede sem fio outros tipos de dispositivos que possibilitam a comunicação pela rede, seja a conexão 3G, 4G, bluetooth ou qualquer outro mecanismo que permita a troca de dados entre os aparelhos.

### 1.1.7. Memória ROM (Read-only Memory ou Memória Somente Leitura)

Lembra quando você liga o computador e aparecem alguns escritos, às vezes alguma imagem, logotipo na tela? Esses tipos de informações estão gravadas em uma memória chamada de “somente leitura”, apesar desse nome ela é gravável, de gravação lenta e dificultada. Essa memória também é responsável por guardar as informações da **BIOS (Basic Input/Output System)** ou Sistema Básico de Entrada/Saída, que é responsável por guardar informações sobre o hardware, possibilitando alguns ajustes básicos, como definir a ordem em que componentes como HD, pendrive, rede, drive de DVD, entre outros irão iniciar. Também grava a data e hora do sistema, exibe a temperatura da CPU e da placa-mãe em alguns modelos.

Em computadores mais recentes, a **BIOS** tem sido substituída pela **EFI (Extensible Firmware Interface)** ou **UEFI (Unified Extensible Firmware Interface)**, que os fabricantes alegam trazer mais recursos e segurança para o uso do computador, porém também cria entraves para a instalação de sistemas operacionais não certificados pelo fabricante da **UEFI**.

### 1.1.8. Memória Permanente

O disco rígido mais tradicional, o **HD**, é um dispositivo de metal que no interior possui discos metálicos, motores, cabeças de gravação e leitura que salvam e leem os dados nesses discos metálicos, isso é feito utilizando magnetismo.

**SSD**, pendrive e cartão de memória são praticamente a mesma tecnologia, mas utilizando encaixes e com capacidades e desempenhos diferentes. A vantagem dessa tecnologia é que todo o processo é feito apenas com corrente elétrica, não necessitando de motores, o que possibilita gravar e ler dados de forma muito mais rápida.

Em todos os casos o objetivo é o mesmo, armazenar os dados a longo prazo e de forma que seja simples incluir novos dados, alterar os já gravados ou apagá-los. O sistema operacional, programas, suas fotos, arquivos de texto, ou seja, tudo que fica armazenado no computador está na memória permanente. Nos capítulos a seguir você vai aprender como acessar e modificar esses dados utilizando comandos no Shell.

### 1.1.9. Memória RAM (Random Access Memory ou Memória de Acesso Aleatório)

O processador faz seu trabalho de forma realmente muito rápida, mas para isso ele precisa ser servido de informações, que ficam armazenadas na memória; primeiro o computador acessa a memória permanente, que acabamos de ler sobre, então uma parte dessa memória é transferida para a memória **RAM**, que possui muito mais velocidade, para evitar que o processador fique ocioso e cause lentidão no sistema. O processador trabalha de forma tão rápida que sequer a memória **RAM** possui velocidade para suprir completamente essa necessidade. No próprio processador existe uma memória de altíssima velocidade, chamada de memória cache, porém, diferente da memória **RAM**, cada processador já vem com sua memória cache embutida e não é possível alterar sua quantidade.

Para termos uma ideia melhor sobre as velocidades, vejamos a seguinte comparação: um **HD** dificilmente lê mais de 100 MB/s e um SSD dos melhores dificilmente chega a 1000 MB/s, memórias **RAM DDR3** ligadas em **Dual Channel** podem facilmente passar de 25000 MB/s.

Acredite, memória **RAM** nunca é demais! Você pode notar o seu computador lento quando abre muitos programas, geralmente a causa é pouca memória **RAM**. Todo e qualquer programa aberto no computador é enviado, armazenado e trabalhado na memória **RAM** e, somente quando ele é fechado, pode ser removido da memória **RAM**. Imagine que você possui 4GB de memória **RAM** e está com apenas 100MB livre, então resolve executar um jogo que ocupa 500MB de memória, isso não será possível. Para que você possa abrir o jogo, será preciso fechar alguns programas que estão abertos para que mais memória seja liberada possibilitando a abertura do seu jogo.

O exemplo anterior sobre o jogo não considerou a possibilidade de uso de memória virtual, também conhecida como swap, porém o seu uso é bastante comum nos sistemas operacionais, ao atingir o limite de uso da memória **RAM**, parte da memória permanente pode ser utilizada para armazenar dados que deveriam estar na memória **RAM**, assim liberando espaço para novos programas serem abertos. Ainda que essa função permita abrir mais programas, a partir do momento que se passa a utilizar parte da memória permanente como auxílio da memória **RAM**, o sistema irá ficar cada vez mais lento, pois, como vimos, a velocidade das memórias permanentes é muitas vezes menor.

Uma característica da memória **RAM** é ser volátil, ou seja, ao encerrar o fornecimento de energia os dados são perdidos, por isso que ao faltar energia o sistema precisa passar por todo o processo de boot, caso contrário bastaria o hardware ser ativado novamente e não precisaríamos esperar o sistema operacional e os aplicativos serem carregados, tudo estaria pronto para uso de forma quase imediata. Todo esse processo de passar dados da memória permanente para a **RAM** e salvar novamente na memória permanente está próximo de se tornar obsoleto, em poucos anos deverá estar popular o memristor, que é um tipo de memória tão rápido quanto a **RAM**, mas que possui tanta capacidade de armazenamento quanto o **HD** e não perde as informações quando o fornecimento de energia é interrompido.

### 1.1.10. Gabinete e Cooler

A quantidade de coolers pode variar, pode não ter nenhum ou até vários, mas o mais comum é que tenha ao menos um para o processador. O **Cooler** é o sistema ativo de refrigeração, em geral um pequeno ventilador, mas existem também os que utilizam tubos com líquidos para refrigerar. É comum que eles também estejam presentes na fonte e na placa de vídeo.

Gabinete é a parte externa, todos os componentes são fixados no seu interior, em geral suas diferenças são estéticas, mas pode haver diferenças na sua capacidade de refrigeração, quantidade de portas **USB** frontais e também é importante que seja um gabinete compatível com o tamanho do equipamento.

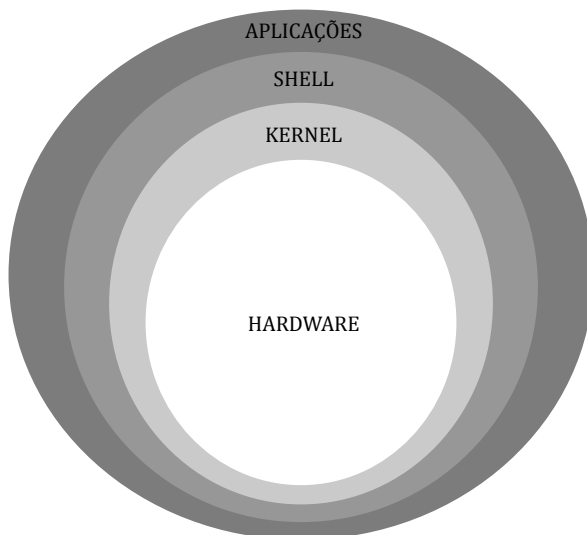
## 1.2. Conhecendo o Ambiente Linux

Podemos separar em camadas a forma que o ambiente **Linux** funciona, na camada mais básica temos o hardware, sem ele nada é possível na informática. Quem faz a comunicação direta com o hardware é o kernel, ele se torna o responsável por controlar o funcionamento do hardware, que gerencia memória, rede, processador, placa de vídeo, de áudio e todas as tarefas básicas para que o ambiente funcione.

O **Linux** em si é apenas o kernel, ao juntá-lo com o shell e os outros programas temos então o ambiente **Linux**. É bastante normal vermos citações sobre esse ambiente como **GNU/Linux**, e isso é correto, pois os programas fornecidos pelo projeto **GNU** são essenciais para o funcionamento do ambiente **Linux**. Neste livro, e na maior parte dos locais, nos referimos ao ambiente apenas como **Linux**, não por desmerecimento ao projeto **GNU**, que é de extrema importância, mas temos de convir que não é uma pronúncia agradável de ficar repetindo, **GNU/Linux**. Portanto lembramos da importância do projeto **GNU**, mas dizemos somente **Linux**, por praticidade.

Por ser habitual o uso de camadas para representar o funcionamento do ambiente, o shell é considerado o acesso de mais baixo nível do sistema, pois é o nível mais próximo do **kernel** e do **hardware**, já os aplicativos que utilizamos a partir do shell estão uma camada acima. Ainda podem existir mais camadas dentro dos próprios aplicativos, cada camada extra se diz que está em um nível mais alto. Nesse caso não se aplica o que é habitual dizer, que algo de alto nível é melhor que algo de baixo nível. Os níveis a que estamos nos referindo existem apenas para dizer por quantos programas um comando tem de passar até chegar ao **kernel** e ao **hardware**, quanto menos programas, mais baixo é o nível de operação.

Acima do **shell**, ou seja, os programas que executamos por meio do **shell**, já são de uma camada superior, nela que estão contidos todos os programas que estamos habituados a utilizar, por exemplo: navegadores e editores de textos. Essa camada também inclui o modo gráfico, que permite que utilizemos interfaces com as mais variadas formas e quantidade de cores, pois o **shell** em si é uma interface somente de texto.



No processo de boot é iniciada a **BIOS/EFI**, que inicia o gerenciador de boot. Apesar de existirem vários boot-loaders (carregadores de boot), o mais utilizado se chama **GRUB**. Este indica a localização do **kernel**, do **initrd** e do **init**, que irão deixar o sistema pronto para o uso. O **kernel** é iniciado e ativa o **initrd** que executa uma série de checagens para assegurar o correto funcionamento do computador. Em algumas distribuições o **initrd** não existe, mas isso não é comum. Ao final dessa checagem é executado o **init**, que se torna responsável pela execução de todos os programas no sistema, também chamado de **processo 1** ou **PID 1**. Nesse momento o **init** prossegue com o início dos programas que estão determinados em sua configuração, grande parte tende a ser para a detecção do hardware e início de programas essenciais, como os que gerenciam a rede. Ao final do boot o sistema deve disponibilizar algumas sessões do **shell**, prontas para efetuar login, mesmo quando o sistema possui modo gráfico instalado.

## 1.3. Como Acessar o Shell

Ao iniciar um computador com **Linux** sem modo gráfico, a primeira interação que temos é com o terminal ou com sua tela de login, porém com o modo gráfico instalado é natural que o sistema inicie automaticamente e ofereça uma tela de login para o ambiente gráfico, nesse caso você pode efetuar login no sistema e procurar por um emulador de terminal.

Podemos acessar o shell pela rede, inclusive, em servidores utilizando **Linux** é habitual não existir modo gráfico. Muitas vezes sequer o servidor está ligado a um monitor, teclado e mouse, nesses casos efetuamos toda a configuração e manutenção do servidor pela rede, por meio de um programa chamado de **SSH (Secure Shell)**. Ele permite efetuar conexões seguras, ou seja, criptografadas, e utilizar de forma remota o terminal do servidor. Dessa forma, caso você pretenda administrar servidores **Linux**, é fundamental aprender sobre **shell**. Mais informações sobre **SSH** serão abordadas no capítulo 5.

### 1.3.1. Principais Shells

O **shell** não é um programa em si, mas uma categoria de programas, existem diversos tipos, e, por serem da mesma categoria, possuem praticamente a mesma finalidade. Assim como podemos escolher qual navegador utilizar para acessar a internet, por exemplo: Firefox ou Google Chrome, o mesmo ocorre com o shell, e podemos optar entre diversas opções:

- **Bash (GNU Bourne Again Shell)**: É o tipo de shell mais utilizado e também o que iremos utilizar como base para o conteúdo deste livro.
- **Dash (Debian Almquist shell)**: Também é bastante utilizado e quase todo o conteúdo deste livro funciona normalmente ao utilizá-lo. Mesmo os outros tipos de shell, em geral, possuem uma forma de operar bastante parecidas.
- **Sh (Bourne Shell)**: Criado por Stephen Bourne na AT&T. Por muitos anos esse shell foi o padrão do sistema **Unix**. Também é chamado de Standard Shell, foi portado para todos os ambientes **Unix** e **Linux**.

- **Ksh (Korn Shell)**: Criado por David Korn. Pode-se dizer que o **Ksh** é uma atualização do **Sh**, pois possui todas as opções do **Sh** e outras funcionalidades. É totalmente compatível com o **Sh**.
- **Csh (C Shell)**: Criado por Bill Joy da universidade de Berkley. É o shell mais utilizado em ambientes **BSD** e **Xenix**. Possui uma estruturação dos comandos bem semelhantes a linguagem **C**, porém não há compatibilidade com o **Sh** (Bourne Shell).

Lembrando que existem outros tipos de **Shell**, porém não tão conhecidos ou utilizados nos ambientes **Linux** ou **Unix**.



**Observação:** Existe uma variável de ambiente que armazena o shell padrão que está sendo utilizado pelo sistema, é a variável **\$SHELL**. O arquivo de configuração **/etc/shells** contém todos os shells que estão instalados no sistema.

---

### 1.3.2. Encontre o Shell

Para começar a praticar o que é ensinado neste livro, que trata do **Shell Linux**, é interessante que você saiba como encontrar um **shell** no **Linux** para início da prática.

Caso você já possua um sistema **Linux** instalado, o mais provável é que esteja utilizando a interface gráfica. Existem caminhos diferentes para acessar o terminal, o modo mais prático é por meio dos emuladores de terminal. Existem diversos emuladores de terminal, por exemplo: **xterm**, **aterm**, **gnome-terminal**, **konsole**, **guake**, dentre outros. Podem ser encontrados no menu de aplicativos do seu ambiente gráfico preferido. Normalmente digitando na busca do menu de aplicativos por "**Terminal**" é apontado para alguns desses que foram informados.

Existe a opção de utilizar diretamente o modo texto, ou seja, fora da interface gráfica. Você poderá acessar o terminal de comandos, **shell**, utilizando um conjunto de teclas a serem pressionadas simultaneamente. Na parte superior do teclado existe uma série de teclas iniciadas com a letra **F** seguida de um número, a combinação deve ser feita pressionando as teclas **<Ctrl+Alt>+(F1-F12)>**, em que cada tecla **F** seguida de um número representa um terminal, uma interface gráfica ou simplesmente uma tela de espera em que pode ser aberto mais um terminal ou uma interface gráfica.

Normalmente ficam à disposição seis terminais, chamadas de **tty (teletypewriter)**, que são selecionados pressionando simultaneamente, por exemplo, as teclas **<Ctrl+Alt+F1>**. Para voltar ao modo gráfico, utilize o mesmo tipo de combinação, porém tente utilizar alguma tecla entre **F7** e **F12**, geralmente o modo gráfico padrão se localiza na combinação **<Ctrl+Alt+F7>**.

Para maior comodidade, é recomendado que utilize uma distribuição **Linux** com modo gráfico e execute os comandos deste livro por meio de um emulador de terminal. Porém caso opte por utilizar direto pela **tty**, lembre-se de que existem outras telas **tty** acessíveis por meio dos atalhos do teclado, isso pode ser bastante útil quando um programa está executando em uma tela, assim podemos utilizar as outras. Quando se utiliza o modo gráfico, basta abrir novas abas no emulador de terminal ou mesmo novas janelas.