Uirá Ribeiro

Linux

Para Desenvolvedores

Copyright © 2020 by Uirá Ribeiro

Todos os Direitos Reservados

Linux para Desenvolvedores

Tudo que você precisa saber de Linux para desenvolver software

Uirá Ribeiro

ISBN 978-65-990965-0-1

CAPÍTULO DE AMOSTRA

Este livro e todo seu conteúdo é protegido pela Lei de Direitos Autorais - LEI Nº 9.610, de 19 de Fevereiro de 1998. Este livro e suas partes **não poderão** ser reproduzidos / distribuídos / copiados / cedidos / vendidos / emprestados sem a autorização prévia e por escrito do autor, sob pena da adoção das medidas cabíveis na esfera cível e penal.

Os conceitos emitidos neste livro são de inteira responsabilidade do Autor.

Apesar de toda a atenção, erros de digitação e/ou de impressão não são descartados. Em caso de alguma dúvida, entre em contato conosco pelo e-mail uribeiro@gmail.com para que possamos ajudá-lo.

O Autor se exclui de quaisquer responsabilidades por eventuais perdas ou danos a pessoas ou bens por uso deste livro.

Sobre o Autor

Se colocarmos todos os sistemas e linguagens em que prof. Uirá Ribeiro é fluente, você terá a certeza de que ele é um Jedi. Vamos deixar você na dúvida dizendo apenas que ele é Mestre em Sistemas distribuídos, especialista em telecomunicações, redes de comutação de pacotes, Voz sobreIP, gateways de acesso remoto, gatekeepers. A sopa de letrinhas você encontra nas buscas on-line.

Se buscar, vai descobrir também que Uirá une duas paixões em suas atividades: TI e educação. A aventura que começou como diretor de TI de nove campi o levou a assumir a Direção geral da Universidade Salgado de Oliveira em Belo Horizonte.

Achando que era pouco, o nosso autor continuou se dedicando ao site www.certificacaolinux.com.br, com vários cursos OpenSource e preparatório para os exames da LPI e CompTIA.

A Força permite que ele durma pouco, de forma que ainda consegue tempo para alimentar o hábito de cozinhar e viver inventando soluções – a última foi uma câmera de segurança criada com um computador de 8 dólares rodando Linux!

Enquanto restaura a paz na república, mantém seus penguins alimentados e saudáveis desde 1998, e aprenderam a voar nas "núvens" já faz tempo.

Também é autor do livro “Sistemas Distribuídos: desenvolvendo aplicações de alta performance no Linux”, “Metodologia Científica: teoria e prática”, “TCC – Trabalho de Conclusão de Curso”, "Certificação Linux", "Certificação Linux para LPIC-101", "Certificação Linux para LPIC-102", "Certificação Linux Essentials", "Certificação CompTIA Linux" e “37 Hábitos dos Professores Altamente Eficazes”, com títulos em português, inglês, espanhol e italiano.

E assim, entre invenções, linguagens e delícias, o prof. Uirá divide com você o conhecimento de "Dev em Ops" nas páginas deste livro.

Sumário

Linux para Desenvolvedores i
Sobre o Autor iii
1 - Escolha do Linux e Instalação de Software 1
A evolução do Linux e sistemas operacionais populares 1
Distribuições Linux 2
Sistemas Embarcados 3
Android 3
Raspberry Pi 3
Linux na Nuvem 4
Escolha do Linux 4
Como Escolher um Sistema Operacional Linux 4
Ciclo de Vida 5
Desktops Gráficos 6
KDE 6
GNOME 8
Xfce 10
Cinnamon 11
Linha de Comando 12
Usos do Linux na Indústria 12
Virtualização em Contêiner 14
DevOps e Contêineres 14
Contêiner são VM's? 14
Software para Conteineres 15
LXC 15
Docker 15
Gerenciamento de Pacotes Debian 15
dpkg 17
apt-get 19
apt 21
Gerenciamento de Arquivos Red Hat e YUM 22
rpm 23
yum 29
dnf 33
zypper 35
Gerenciando Repositórios 35
Repositórios no Debian/Ubuntu 36
Adicionando um repositório 37
Repositórios no Red Hat/CentOS (Yum) 38
Grupos de Pacotes 39
2 - Bases da Linha de Comando 43
Tudo nasceu no Shell 43
A Linha de Comandos 43

Variáveis de Shell 45
Uso de Coringas 48
history 49
script 51
type 52
echo 53
Como obter ajuda na linha de comando 53
Help dos Comandos 54
Utilizando as Páginas Info 54
Encontrando a Documentação em /usr/share/doc 55
man 55
locate 57
find 58
Como usar diretórios e listar arquivos 60
/ 61
/bin 62
/boot 62
/dev 62
/etc 62
/home 62
/lib 63
/mnt 63
/media 63
/proc 63
/opt 63
/root 63
/run 63
/sbin 63
/tmp 63
/usr 64
/var 64
Detalhes sobre os Arquivos e Diretórios 65
Arquivos e Diretórios Escondidos 65
Diretório Corrente e Diretório Anterior 66
Diretório Home 67
ls 68
pwd 70
Ir e Voltar nos Diretórios 71
df 72
du 73
alias 75
whereis 76
which 77
3 - Criar, Mover e Remover Arquivos 79
Comandos básicos com arquivos e diretórios 79

touch 79
cp 81
mv 82
rm 83
mkdir 84
rmdir 84
4 - Montando e Desmontando Sistemas de Arquivos 86
Hierarquia de Diretórios e não letrinhas 86
mount 88
/etc/mtab 90
/proc/partitions 90
umount 91
lsblk 93
blkid 94
/etc/fstab 95
5 - Backup e Arquivamento 98
Como comprimir arquivos com a linha de comando 98
gzip e gunzip 98
bzip2 e bunzip2 99
xz e unxz 99
Arquivos tarball 100
6 - Como buscar e extrair dados de arquivos 104
Redirecionamento de Dados 104
cat 109
less 110
head 111
tail 111
Multitail 112
sort 112
uniq 114
wc 115
cut 116
tee 117
column 117
printf 120
Procurar por Arquivos Texto Usando Expressões Regulares 120
Conceitos Básicos 121
Alternância 121
Agrupamento 121
Quantificação (ou repetição) 121
POSIX 121
BRE: expressões regulares básicas 122
ERE: expressões regulares estendidas 123

Classes de caracteres 123
grep 124
egrep 128
fgrep 129
xargs 129
7 - Transformação de Dados 132
Dividir, alterar e moldar arquivos 132
tr 132
split 134
awk 135
sed 137
diff 140
jq 140
8 - Cópia Remota de Arquivos 143
Porque 143
scp 143
rsync 144
sftp 147
curl 148
Formatando o JSON com jq 150
httpie 152
lynx 154
9 - Localização 156
...num país tropical... 156
Configurando as Zonas de Fuso Horário 156
/usr/share/zoneinfo 156
/etc/localtime 157
/etc/timezone 157
Variável TZ 158
tzselect 158
date 160
hwclock 162
timedatectl 163
Variáveis de Localização 164
locale 165
localectl 166
Padrões de Caractere 167
ASCII 167
ISO-8859 168
..... 169
UNICODE 169
iconv 170
10 - Gerenciamento de Serviços 172

O Gerenciador de Tudo 172
Systemd 172
Modo de Emergência e Resgate 174
Rescue.target 174
Emergency.target 175
systemctl 175
Desabilitando Serviços não Necessários 179
Super Servidores 180
/etc/xinetd.conf 181
/etc/xinetd.d 181
Uso de Tcpwrappers 182
hosts.allow e hosts.deny 182
Alterando as Portas Padrão 183
Restringindo acesso ao Cront e At 183
11 - Acesso Remoto 185
Criptografia de Dados 185
Par de Chaves de Criptografia 185
Public Key Infrastructure - PKI 186
Auto-assinatura de Certificado Digital 187
Hashing 187
Assinando Transmissões com Hashing 188
Secure Shell - SSH 188
Algoritmos de Criptografia 189
Configuração do cliente OpenSSH 2 189
Autenticação sem senha 190
/etc/ssh_config 190
Conexão Remota usando o SSH 190
ssh_know_hosts 191
OpenSSH Server 191
sshd_config 192
Habiltar o sshd 192
ssh-agent e ssh-add 193
ssh-copy-id 194
Deixando o SSHD mais Seguro 194
Trocar a porta do SSH 194
Não permitir login com o Root 194
Uso de Tcpwrapper 195
Cliente VPN 196
SSL/TLS 197
DTLS 197
IPSec 197
Instalação do Cliente VPN 198
Multiplexação de Terminais 199
screen 200
tmux 201
12 - Automatizar e Agendar Trabalhos 203

Como Agendar Trabalhos no Linux 203
cron 203
crontab 205
Permissões do Cron 206
/etc/cron.allow 206
/etc/cron.deny 206
Diretórios Especiais do Cron 207
/etc/cron.d 207
/etc/cron.{daily, hourly, weekly, monthly} 207
at 208
atq 210
atrm 210
Permissões do at 210
Systemd Timer 210
Unidades de Tempo Passageiras (Transientes) 213
systemd-run 213
13 - Shell Script 215
Como transformar comandos em script 215
Variáveis do Shell 215
Variáveis de Ambiente ou Globais 215
PATH 215
HOME 215
USER 215
TERM 216
Variáveis locais 217
Alterando as Variáveis de Shell 218
Escolhendo o Interpretador de Script #! shebang 220
Permissões de Execução 221
Variáveis Especiais 222
Expansões de Shell 224
export 226
set 226
unset 228
env 229
source 230
printenv 231
O Comando test 232
Uso de [ ] no lugar do "test" 234
Operadores || e && 234
if 235
Recebendo Dados dos Usuários com read 235
case 236
for 238
while 239

until 240
autoexpect 241
14 - Compilando Software 244
Compilação de Software 244
Compiladores 244
GCC 244
PRÉ-PROCESSAMENTO 245
COMPILAÇÃO 245
ASSEMBLER 246
LINKER 246
Baixando o Código Fonte 248
wget 248
cURL 249
Dados da Arquitetura 249
Compilando Software 250
configure 250
Makefile 251
make 251
Gerenciamento de Bibliotecas Compartilhadas 252
ldd 254
ldconfig 255
LD_LIBRARY_PATH 256
Controle de Versão com Git 256
Novo Projeto com o Git 257
Clonar um Projeto no Github 258
Adicionar arquivos ao Index 259
Adicionar Arquivos ao Repositório Local 259
Visualizando o Log de Alterações 260
Enviando do Repositório Local para o Repositório Remoto 260
Atualizando o Repositório Local 261
Criando um Branch 261
Mesclando um Branch 262
Debug de programas com gdb 263
Utilizando o depurador gdb pela primeira vez 263
Exemplo de depuração de um programa 266
16 - Editor de Textos 274
Editor de Textos vi/vim 274
Truques de Programadores com Vi 276
Colorir a sintaxe 277
Enumerar as linhas 277
Depurador de Sintaxe 278
Undo infinito 279
Trocar texto (substituir) 280
Indentação 281
Retirar os ^M no final dos arquivos 282

vimdiff 282
nano 283
17 - Processos no Linux 285
Noções básicas sobre o hardware no Linux 285
Gerência de Dispositivos 285
Armazenamento 286
Informações de Hardware no Linux 287
Processos no Linux 287
O que é um processo? 288
ps 289
Usuário dono do processo (UID) 293
Número de Processo - Process ID (PID) 293
Processo Pai (PPID) 293
% de CPU 293
% de Memória 293
Hora de início (STIME) 294
Tempo de CPU (TIME) 294
Linha de comando (CMD) 294
Terminal (TTY) 294
top 295
free 297
uptime 298
Sinais 299
kill 300
killall 301
pgrep 302
pkill 303
Multitarefa: processos em segundo plano e processos em primeiro plano 303
Processos em primeiro plano (Foreground) 303
Processos em segundo plano (Background) 304
& 304
Controle de Processos 305
bg 305
fg 306
jobs 307
nohup 307
watch 308
18 - Análise de Log 310
Serviços de LOG 310
rsyslog 310
Rodízio de Logs com Logrotate 311
Journal do Systemd 312
journalctl 312
dmesg 313

19 - Networking 315
Seu computador na rede 315
Localhost 315
Redes Privadas 316
Protocolos que Compõem o TCP/IP 316
Camada de Transporte 316
TCP - Transmission Control Protocol 317
UDP - User Datagram Protocol 317
IP - Internet Protocol 317
ICMP - Internet Control Message Protocol, ... 317
Serviços e Portas TCP/IP 318
Tabelas de Rotas 319
IPv6 319
/etc/hostname 321
/etc/hosts 321
/etc/resolv.conf 321
/etc/host.conf 322
/etc/nsswitch.conf 322
ip 323
ping e ping6 326
ss 327
Configurar o Cliente DNS 329
host 331
route 332
netstat 333
traceroute e traceroute6 335
tracepath e tracepath6 336
Portas Abertas com netstat, ss e lsof 337
lsof 338
nmap 338
netcat 339
iftop 340
fuser 341
dig 342
Acesso Remoto Desktop 343
X11 343
XDMCP 343
VNC 343
SPICE 344
RDP 344
NX 344
20 - Criando usuários e grupos 346
Gerenciamento de Usuários 346
useradd 346

Diretório Padrão (skel) 347
usermod 348
userdel 349
groupadd 350
groupdel 350
passwd 351
Arquivos de Configuração do Bash e Perfil 353
/etc/profile 353
/etc/profile.d 353
/etc/bashrc ou /etc/bash.bashrc 353
~/.bash_profile 353
~/.bash_login 354
~/.profile 354
~/.bashrc 354
~/.bash_logout 354
Ajustando o Ambiente de Trabalho dos Usuários 356
21 - Permissões e Propriedade de Arquivos 357
Gerenciando Permissões de Arquivo e Propriedade 357
Você não precisa decorar a tabela de permissões 361
Modificando as Permissões de Arquivos 362
chmod 362
Alterando o Dono dos Arquivos e Grupos 365
chown 365
chgrp 367
22 - Segurança Básica e Permissões Especiais 368
Segurança básica e identificação de tipos de usuário 368
/etc/passwd 368
/etc/shadow 369
/etc/group 370
passwd 371
Identificando Qual Usuário Está Conectado ou Última Conexão 371
last 371
id 372
whoami 372
who 373
w 373
Executando Programas com a Permissão de Outro Usuário 373
sudo 374
su 374
sudoedit 375
23 - Monitoramento de Processos 378
Monitoramento e Configuração de CPU 378
uptime 379

sysstat 381
Monitoramento e Configuração de Memória 382
Swap 383
mkswap 383
htop 385
time 386
lsof 386
strace 389
24 - Diretórios e Arquivos Especiais 392
Sempre precisamos de um 392
Arquivos temporários 392
/tmp 392
/var/tmp 392
/run 392
Links Simbólicos e Hard Links 393
Links Simbólicos 393
Hard Links ou Links Físicos 393
ln 393
Perda de Senha de Root 395
Outros Livros do Autor 397
Índice Remissívo 403

16 - Editor de Textos

"Eu sei como correr sem você segurando minha mão!"
-- Rey

Editor de Textos vi/vim

$ vi [opções] arquivo

O vi é um editor de textos ASCII poderoso e muito usado na interface de caractere do Linux para edição de arquivos e programas. Seu uso não é muito intuitivo à primeira vista, mas a edição simples de textos pode ser feita usando poucos comandos.

Ele pode ser muito útil para pequenas correções pontuais, principalmente em ambiente de produção. Então não é má idéia saber lidar com esse editor de textos universalmente disponível nas distribuições Linux.

Basicamente ele tem dois modos de trabalho: o modo de operação e o modo de inserção. No modo de operação o vi espera comandos que vão realizar alguma ação. No modo de inserção, tudo que for digitado é considerado texto.

Figura 20 – Vi

O modo de operação é acessado através da tecla ESC. O vi sempre inicia nesse modo. Para entrar no modo de inserção, basta a pressionar a tecla INSERT ou o comando “i”.

Para fazer a edição simples de arquivos no vi:

```
$ vi nomedoarquivo
```

Altere o modo para inserção com a tecla INSERT ou pressionando a tecla "**i**". O cursor pode ser movido neste modo com as setas de posição. Ao terminar a digitação aperte a tecla ESC e digite:

> :w para salvar as modificações;
> :wq para salvar e sair do vi;
> :q! para sair sem salvar;

Veja a seguir a tabela de comandos aceitos pelo vi. Todos os comandos deverão ser dados no modo de operação. O sinal “+” indica teclas que devem ser pressionadas simultaneamente. A vírgula indica que as teclas devem ser digitadas em sequência.

Ação	Comando
Abrir Arquivo	:e arquivo
Salvar Arquivo	:w
Salvar Arquivo Como	:w arquivo
Salvar e Sair	:wq
Sair sem salvar	:q!
Gravar conteúdo se alterado	:ZZ
Para marcar um texto para cópia ou corte	v, setas de direção
Para copiar texto marcado	y
Para cortar texto marcado	c
Para colar texto marcado	p
Copiar uma linha	yy
Copiar até o final do arquivo	yG
Apagar texto à frente (DEL)	x
Apagar texto para trás (BACKSPACE)	X
Apagar uma linha	dd
Apagar até o final do arquivo	dG
Apagar até o final da linha	D

Ação	Comando
Localizar texto à frente	/texto
Localizar novamente	/
Localizar texto para trás	?texto
Localizar novamente	?
Desfazer alterações	u
Refazer alterações	CTRL+r
Formatar Alinhamento Centralizado	:ce
Alinhamento à direta	:ri
Alinhamento à esquerda	:le
Abrir nova janela	:new
Dividir a janela atual em duas	:split
Abrir arquivo em nova janela	:split arquivo
Ir para janela de cima	CTRL+w, k
Ir para janela de baixo	CTRL+w, j
Ir para o início da linha	0
Ir para o final da linha	$
Ir para o final do arquivo	G
Ir para a linha de baixo	j
Ir para a linha de cima	k
Ir para a esquerda	h
Ir para a direita	l
Ir para o final da tela	L
Ajuda	:help
Insere texto antes do cursor	i
Insere texto depois do cursor	a
Insere texto em uma nova linha	o

Truques de Programadores com Vi

O vi ao ser executado procura por um arquivo de configuração chamado "**.vimrc**" no

diretório HOME dos usuários. Nem sempre este arquivo existe, mas pode ser criado pelo usuário para fazer algumas coisas bacanas com o vi, tais como:

Colorir a sintaxe

Para colorir a sintaxe de arquivos com código fonte, deve-se habilitar a diretiva "sintax on" no arquivo .vimrc, com o comando:

```
$ echo 'sintax on' >> ~/.vimrc
```

Desta forma, o vi irá colorir o texto, reconhecendo diversas linguagens de programação. Utilize as aspas simples nestes comandos, como demonstrado.

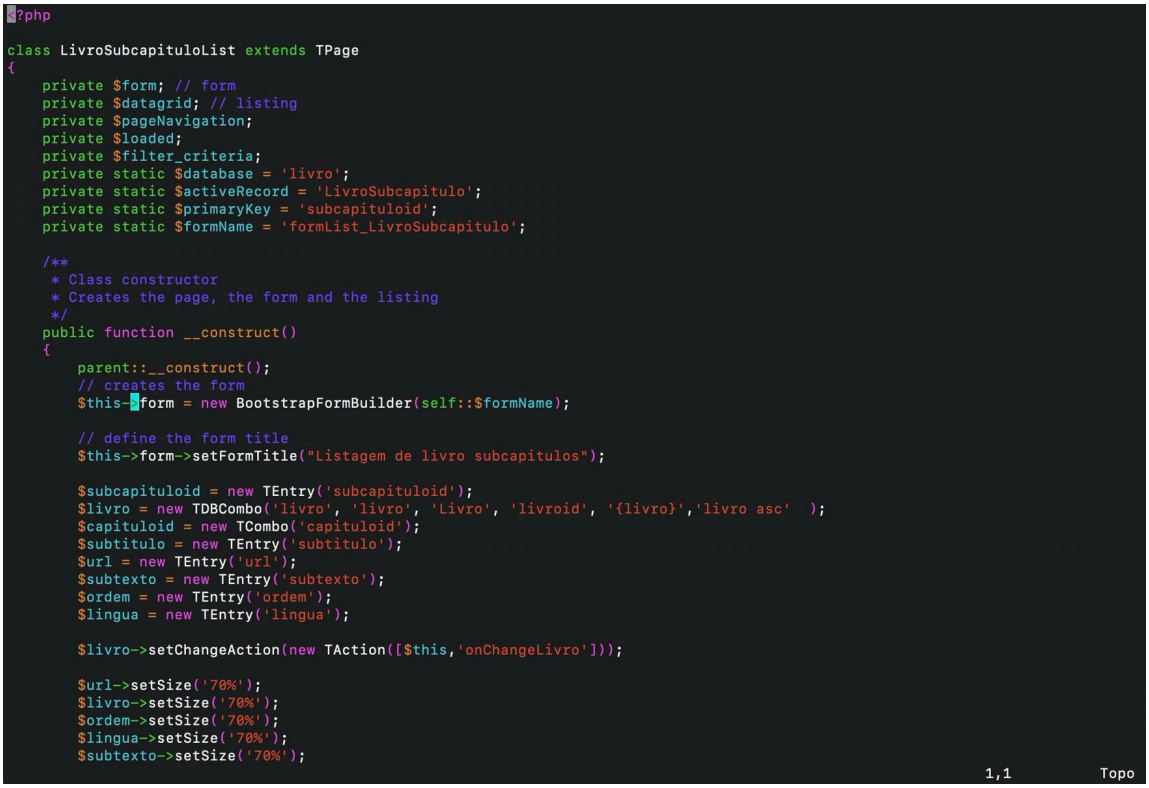


Figura 21 – Sintaxe on no vim

Enumerar as linhas

Outro recursos interessante no vim é a enumeração das linhas, que permite debugar facilmente um código fonte. Para habilitar esse recurso deve-se usar a diretiva "set number" no arquivo .vimrc, com o comando:

```
$ echo 'set number' >> ~/.vimrc
```

Desta forma, o vi irá enumerar o texto, facilitando a depuração do código.

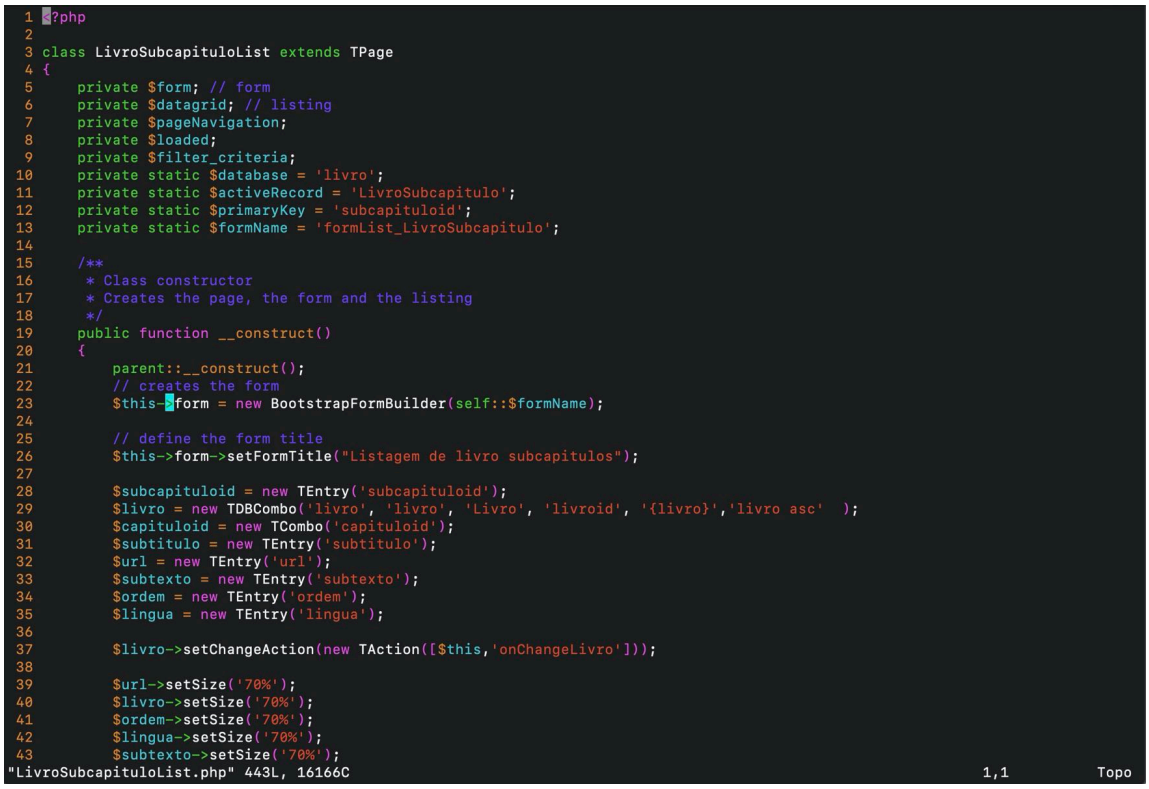


Figura 22 – Enumerar as linhas

Depurador de Sintaxe

Esta opção do vi é muito interessante para os programadores. Ela permite que uma tecla seja mapeada para acionar um compilador ou interpretador de linguagem, com o objetivo de se verificar a sintaxe, sem a necessidade de sair do editor de textos.

No exemplo, utiliza-se a opção "-l" do php, que faz somente uma checagem de sintaxe. A combinação de teclas escolhida foi o CTRL B, que executa o comando **php -l** no arquivo fonte.

```
$ echo 'map <C-B> :!php -l %<CR>' >>~/.vimrc
```

Desta forma, para verificar a sintaxe de um código php, basta apertar o CTRL B, que o interpretador php é acionado para verificar a sintaxe:

Figura 23 – Código php com erro

Neste exemplo, induzi um erro de falta de ";". Ao pressionar CTRL B, o php dirá que há um erro de sintaxe:

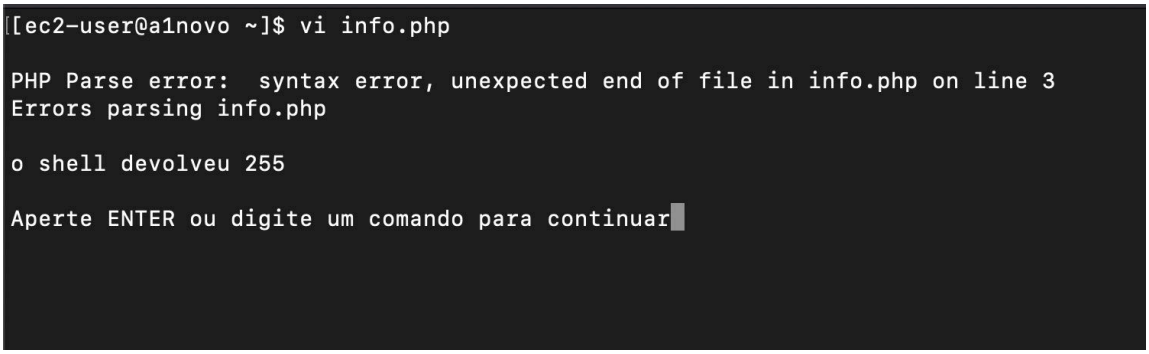


Figura 24 – Chamada ao interpretador

Esta facilidade do vim é extremamente útil em linguagens interpretadas, que permitem verificar a sintaxe sem a necessidade de abandonar o editor.

Para outras linguagens, pode-se usar o mesmo conceito, alterando a opção de checagem de sintaxe:

- PERL: **echo 'map <C-P> :!perl -c %<CR>' >>~/.vimrc**
- PYTHON: **echo 'map <C-Y> :!python -d %<CR>' >>~/.vimrc**

Undo infinito

A tecla "u" no modo de comandos do **vi** permite voltar ao texto anterior, desfazendo as alterações. No entanto, ao sair do vi, a capacidade de desfazer alterações é perdida. Assim, se você estiver editando um código fonte, e sair do editor, não será mais capaz de usar o recurso de desfazer alterações.

Mas há uma solução para o **vi** manter um histórico de alterações, considerada um verdadeiro salvador de vidas.

Ele permite que o **vi** mantenha um histórico de alterações de arquivos em um diretório especial no HOME do usuário.

Para habilitar esse recurso, deve-se executar os seguintes comandos:

Primeiro cria-se o diretório para armazenar o histórico no HOME do usuário:

```
$ mkdir -p ~/.vim/undodir
```

Depois deve-se habilitar o recurso de **undofile**:

```
$ echo 'set undofile' >>~/.vimrc
```

Por último, configura-se o diretório para armazenar esses arquivos, criado no primeiro passo:

```
$echo 'set undodir=~/.vim/undodir' >>~/.vimrc
```

Desta forma, o vi irá manter o histórico de alterações, possibilitando desfazer alterações com a tecla "u".

De vez em quando, é interessante limpar o conteúdo do diretório ~/.vim/undodir.

Trocar texto (substituir)

O recurso de substituir um texto por outro no vim também ajuda muito para fazer substituições de código no vi.

Para fazer uma substituição, entre no modo de comando com a tecla ESC e depois digite:

```
:%s/texto a ser procurado/texto a ser substituído/g
```

Desta forma, o vim irá procurar por todas as ocorrências do **texto a ser procurado** e irá trocar por **texto a ser substituído** no documento todo.

Se você quiser que o vi apenas substitua a primeira ocorrência, o comando é o mesmo, mas sem o "g" no final:

```
:%s/texto a ser procurado/texto a ser substituído/
```

Se o texto a ser procurado ou substituído conter uma barra simples "**/**", deve-se usar uma contra-barra "****" antes, para que o vi saiba que a "**/**" no texto não seja interpretada como um separador.

Neste exemplo, a ocorrência de "**/etc**" será substituída por "**/var**":

```
:%s/\/etc/\/var/g
```

Outra substituição interessante é colocar todas as linhas dentro de uma tag HTML, como por exemplo, de parágrafo:

```
:s/.*/<p>\r&\r<\/p>/
```

A expressão **.*** indica o início da linha, e o "**\r**" o "return" (final da linha).

Procurar texto

Para procurar um texto no vi, basta apertar a barra "/" no modo de comando e escrever o texto. Desta forma, o vi irá procurar as ocorrências do termo buscado uma a uma, pressionando novamente a tecla "/".

Um recurso para turbinar as buscas de texto no vi é habilitar o destaque do termo, bem como habilitar o "case insensitive", possibilitando buscas ignorando se o texto está em caixa alta ou baixa:

Para habilitar o destaque de texto buscado:

```
$ echo 'set hlsearch' >> ~/.vimrc
```

Para ignorar se o termo está em caixa alta ou baixa:

```
$ echo 'set ignorecase' >> ~/.vimrc
```

Para buscar um texto obedecendo a caixa alta, se o termo conter uma letra em maiúsculo:

```
echo 'set smartcase' >> ~/.vimrc
```

Indentação

Um truque interessante do vi é a indentação automática do texto.

Para arrumar um código fonte com indentação, digite no modo de comando:

```
gg=G
```

Desta forma, o vi irá formatar o código, tornando-o mais legível.

Também é possível determinar que o vi indente o código enquanto você programa, variando o espaçamento dependendo do tipo de código fonte.

Para habilitar esse recurso:

```
$ echo 'filetype indent plugin on'  >> ~/.vimrc
```

Depois, deve-se criar um diretório chamado ".vim" no diretório HOME, se esse ainda não existir:

```
$ mkdir ~/.vim
```

Neste diretório, deve-se criar um arquivo para cada tipo de código fonte, como no exemplo:

```
$ cat ~/.vim/php.vim
setlocal shiftwidth=2
setlocal tabstop=2
```

Neste exemplo, há um arquivo php.vim dentro do diretório ~/.vim.

E nele há duas diretivas:

- **shiftwidth=2** Determina que o número de espaços para auto indentação sejam 2. O padrão são 8 espaços.
- **tabstop=2** Determina o número que espaços que o TAB contém. O padrão são 8 espaços.

Para python por exemplo:

```
$ cat ~/.vim/python.vim
setlocal expandtab
setlocal shiftwidth=4
setlocal softtabstop=4
```

Neste exemplo, as diretivas expandtab e softtabstop são para:

- **expandtab** Não faz a troca de TAB por espaços;
- **softtabstop** permite fingir que os TABS são espaços na hora da edição.

Retirar os ^M no final dos arquivos

Geralmente depois de copiar um arquivo texto do Windows para o Linux e editá-lo no vi, o arquivo aparece com códigos "**^M**" no final de cada linha.

Desta forma, pode-se fazer o ajuste do arquivo no vi, utilizando o comando abaixo, no modo de comandos:

```
:e ++ff=dos
```

O comando de substituição também funciona bem:

```
:%s/<Ctrl-V><Ctrl-M>/\r/g
```

vimdiff

$ vimdiff arquivo arquivo2

Uma ferramenta pouco conhecida por programadores, que pode ajudar muito a

encontrar problemas no código é o vimdiff.

Essa ferramenta permite abrir dois arquivos com código-fonte ao mesmo tempo, e faz comparações, indicando em qual linha e o que está diferente entre os arquivos.

Isso é muito útil para verificar diferenças entre um código de desenvolvimento, e outro em produção, especialmente em linguagens interpretadas, arquivos JSON, XML, etc.

Para utilizar o vimdiff, o comando é simples: basta indicar quais são os dois arquivos a serem comparados:

```
$ vimdiff arquivo1.php arquivo2.php
```

O resultado é uma tela com duas janelas, e com destaque das linhas diferentes:

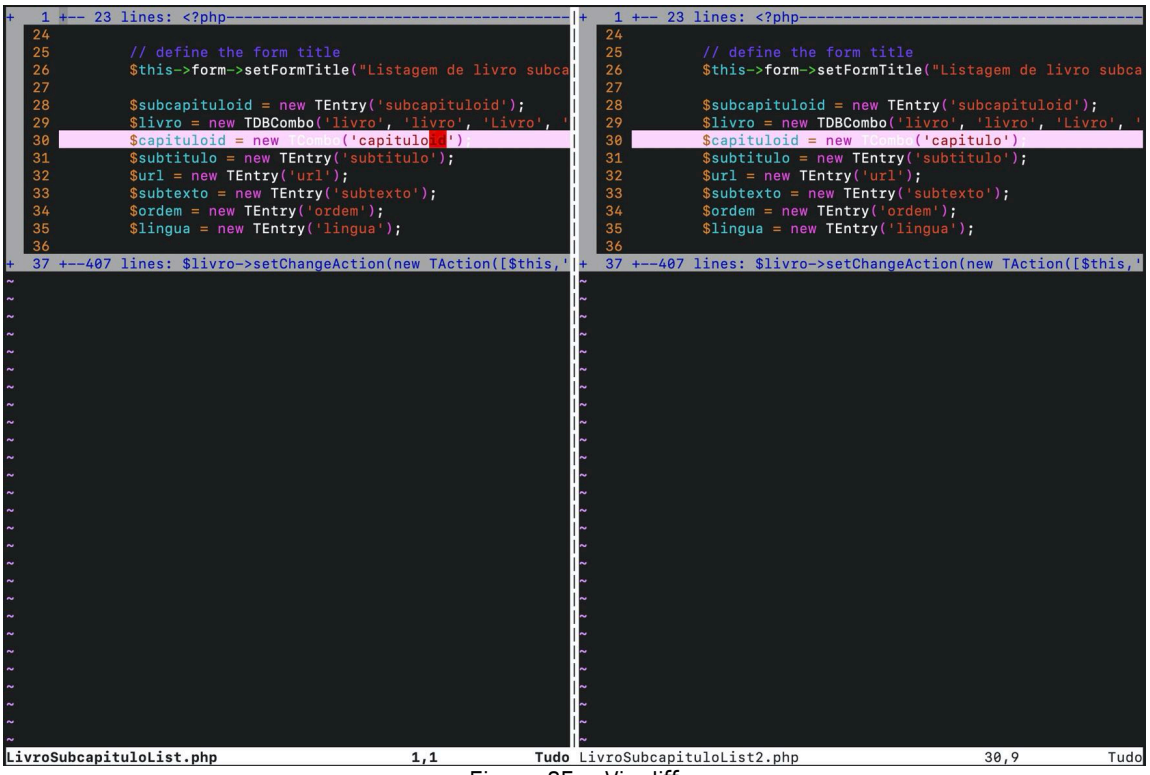


Figura 25 – Vimdiff

nano

$ nano arquivo

Outro editor muito utilizado no Linux, principalmente para iniciantes, é o nano. Ele lembra um pouco o bloco de notas do Windows, mas para terminal do Linux.

O nano é um editor pequeno e amigável. Ele não tem modo de comando. Tudo é simples como editar textos no Bloco de Notas do Windows.

Para executar o nano:

```
$ nano arquivo
```

Os comandos do nano são bem simples, e podem ser acionados com a tecla Control mais a letra correspondente.

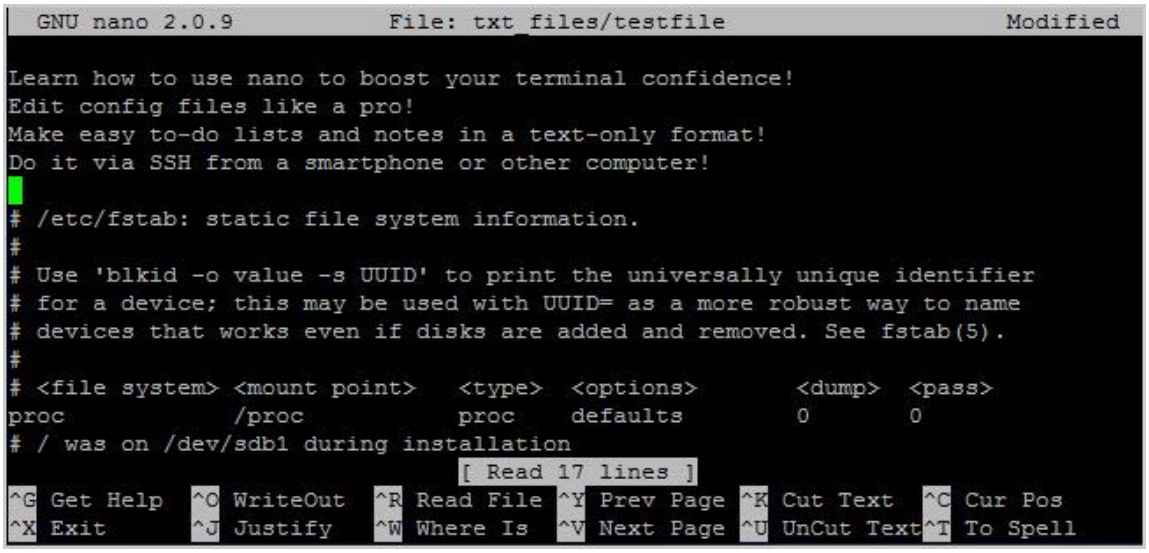


Figura 26 – Nano

"É uma armadilha!"
-- Almirante Ackbar

Outros Livros do Autor

Certificação Linux LPIC-1

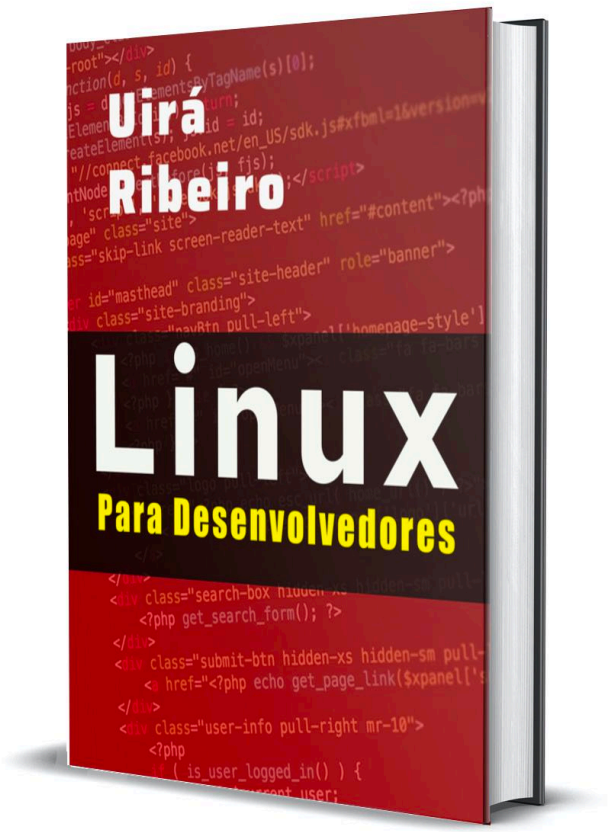
Livro para a Prova LPIC–101

Livro para a Prova LPIC–102

Certificação Red Hat

Livro para Certificação Red Hat EX–200

Linux para Developers

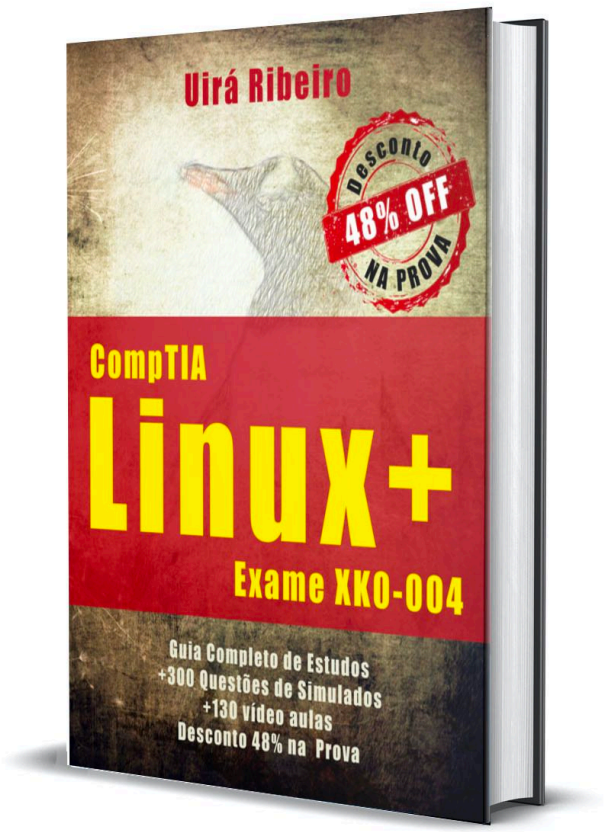


Livro Desenvolvedores

Certificação Linux Essentials

Livro para a Certificação LPI Linux Essentials

Certificação CompTIA Linux+ XK0-004



Livro para a Prova CompTIA Linux+

"Esperança"
-- Leia

Índice Remissívo

A

Acesso Remoto Desktop 343
A evolução do Linux e sistemas operacionais populares 1
Ajustando o Ambiente de Trabalho dos Usuários 356
alias 75
Alterando o Dono dos Arquivos e Grupos 365
apt-get 19
Arquivos de Configuração do Bash e Perfil 353
Arquivos tarball 100
at 208
autoexpect 241
awk 134

B

bg 305
blkid 94
bzip2 e bunzip2 98

C

case 236
cat 109
chgrp 367
chmod 362
chown 365
Cliente VPN 196
column 117
Comandos básicos com arquivos e diretórios 79
Como Agendar Trabalhos no Linux 203
Como comprimir arquivos com a linha de comando 98
Como obter ajuda na linha de comando 53
Como transformar comandos em script 215
Como usar diretórios e listar arquivos 60
Compilação de Software 244
Compiladores 244
Compilando Software 250
Configurar o Cliente DNS 329
Controle de Versão com Git 256
cp 81
Criptografia de Dados 185
cron 203
curl 148
cut 116

D

date 160
Debug de programas com gdb 263
Desabilitando Serviços não Necessários 179
Detalhes sobre os Arquivos e Diretórios 65
df 72
diff 140
dig 342
Diretório Padrão (skel) 347
Dividir, alterar e moldar arquivos 132
dmesg 313
dnf 33
dpkg 17
du 73

E

echo 53
Editor de Textos vi/vim 274
egrep 128
env 229
Escolha do Linux 4
Escolhendo o Interpretador de Script #! shebang 220

/

/etc/fstab 95
/etc/localtime 157

E

Executando Programas com a Permissão de Outro Usuário 373

Expansões de Shell 224
export 226

F

fg 306
fgrep 128
find 58
for 238
Formatando o JSON com jq 150
free 296
fuser 341

G

Gerenciamento de Arquivos Red Hat e YUM 22
Gerenciamento de Bibliotecas Compartilhadas 252
Gerenciamento de Pacotes Debian 15
Gerenciamento de Usuários 346
Gerenciando Permissões de Arquivo e Propriedade 357
Gerenciando Repositórios 35
grep 124
groupadd 350
groupdel 350
Grupos de Pacotes 39
gzip e gunzip 98

H

head 111
Hierarquia de Diretórios e não letrinhas 86
history 49
host 331
htop 385
httpie 152
hwclock 162

I

iconv 170
Identificando Qual Usuário Está Conectado ou Última Conexão 371
if 235
iftop 340
ip 323

J

jobs 306
Journal do Systemd 312
jq 140

K

kill 300
killall 301

L

ldconfig 255
ldd 254
LD_LIBRARY_PATH 255
less 110
Links Simbólicos e Hard Links 393
ln 393
locate 57
ls 68
lsblk 93
lsof 338, 386
lynx 154

M

man 55
mkdir 84
mkswap 383
Modificando as Permissões de Arquivos 362
Monitoramento e Configuração de CPU 378
Monitoramento e Configuração de Memória 382
mount 87
Multiplexação de Terminais 199
Multitarefa
- processos em segundo plano e processos em primeiro plano 303
mv 82

N

nano 283
netcat 339
netstat 333
nmap 338
Noções básicas sobre o hardware no

Linux 285
nohup 307

.

...num país tropical... 156

O

O Comando test 232
O Gerenciador de Tudo 172
Outros Livros do Autor 396

P

Padrões de Caractere 167
passwd 351, 371
Perda de Senha de Root 395
Permissões de Execução 221
pgrep 302
ping e ping6 326
pkill 303
Porque 143
Portas Abertas com netstat, ss e lsof 337
printenv 231
printf 120
Processos no Linux 287
Procurar por Arquivos Texto Usando Expressões Regulares 120
ps 289
pwd 70

R

Recebendo Dados dos Usuários com read 235
Redirecionamento de Dados 104
rm 83
rmdir 84
Rodízio de Logs com Logrotate 311
route 332
rpm 23
rsync 144

S

scp 143
screen 200
script 51
Secure Shell - SSH 188
sed 136
Segurança básica e identificação de tipos de usuário 368
Sempre precisamos de um 392
Serviços de LOG 310
set 226
Seu computador na rede 315
sftp 147
Sinais 299
sort 112
source 230
split 134
ss 327
strace 389
su 374
sudo 374
sudoedit 375
Swap 383
sysstat 381
systemctl 175
Systemd 172
Systemd Timer 210

T

tail 111
tee 117
time 386
timedatectl 163
tmux 201
top 295
touch 79
tr 132
tracepath e tracepath6 336
traceroute e traceroute6 335
Tudo nasceu no Shell 43
type 52
tzselect 158

U

umount 91
uniq 114
unset 228
until 240
uptime 298, 379
useradd 346
userdel 349
usermod 348

V

Variáveis de Localização 164
Variáveis do Shell 215
Variáveis Especiais 222
vimdiff 282

W

w 373
watch 308
wc 115
whereis 76
which 77
while 239
who 373

X

xargs 129
xz e unxz 99

Y

yum 29

Z

zypper 35