

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/309219831>

# MAC OS X: Comandos Básicos e Avançados

Book · June 2016

---

CITATIONS

0

READS

1,541

3 authors:



Alessandro Vivas Andrade

Universidade Federal dos Vales do Jequitinhonha e Mucuri

47 PUBLICATIONS 54 CITATIONS

[SEE PROFILE](#)



Luciana Pereira de Assis

Universidade Federal dos Vales do Jequitinhonha e Mucuri

40 PUBLICATIONS 52 CITATIONS

[SEE PROFILE](#)



André L. Maravilha

Federal University of Minas Gerais

10 PUBLICATIONS 16 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



algoritmo CBA [View project](#)



Informatização dos Processos de Coleta e Análise de Dados de Pesquisa de Demanda Turística [View project](#)



# MAC OS X

comandos básicos e avançados

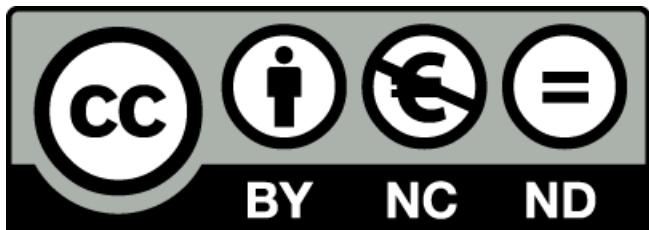
---



Alessandro Vivas Andrade, Luciana P. Assis e André L. Maravilha

## MAC OS X: COMANDOS BÁSICOS E AVANÇADOS

Diamantina  
Alessandro Vivas Andrade  
2016



**EDITOR:** Alessandro Vivas Andrade

**PROJETO GRÁFICO:** Alessandro Vivas Andrade

**CAPA:** Produzida por André L. Maravilha

**Dados Internacionais de Catalogação na Publicação (CIP)**

(eDOC BRASIL, Belo Horizonte/MG)

A553m

Andrade, Alessandro Vivas.

MAC OS X [recurso eletrônico]: comandos básicos e avançados /  
Alessandro Vivas Andrade, Luciana P. Assis, André L. Maravilha. –  
Diamantina (MG): A. V. Andrade, 2016.

171 p. : il. ; PDF

Inclui bibliografia.

Requisitos do sistema: Adobe Acrobat Reader  
ISBN 978-85-920329-1-3

1. MAC OS (Sistema operacional de computador). 2. Macintosh  
(Computador) - Programação. 3. Sistemas operacionais  
(Computadores). I. Assis, Luciana P. II. Maravilha, André L. III. Título.

CDD-005.43

Agência Brasileira do ISBN

ISBN 978-85-920329-1-3

9 788592 032913

# Prefácio

No dia 20 de novembro de 2015 lançamos o livro Linux: Comandos Básicos e Avançados no sítio <http://www.andarilho.pro.br>. O livro foi lançado gratuitamente e em apenas um dia tivemos mais de 1000 downloads oriundos de diversos países do mundo. Atualmente nosso livro conta com mais de 3000 downloads. .

Não temos grandes pretensões desde nosso primeiro projeto. Nossa interesse é sistematizar todo o conhecimento acumulado durante nosso processo de estudo (Graduação, Mestrado e Doutorado) e durante nossa trajetória profissional. O intuito é simples: produzir e disponibilizar conteúdo de qualidade gratuitamente. Não possuímos nenhum tipo de financiamento e arcamos com todo o custo de produção e disponibilização do material.

Recebemos várias perguntas sobre a compatibilidade entre *MAC OS X* e o *Linux* e achamos que a melhor forma de resposta seria uma publicação neste sentido. Animados com a repercussão do primeiro livro começamos a idealizar o nosso segundo projeto que seria uma continuação do primeiro projeto.

Neste livro trabalhamos exclusivamente com comandos que funcionam no **MAC OS X**. Alguns destes comandos funcionam da mesma maneira nos dois sistemas operacionais. Trabalhamos exaustivamente para testar a compatibilidade destes comandos e todos apresentados aqui funcionam no **MAC OS X** versão **El Capitan**.

No início de cada capítulo inserimos previsões realizadas por grandes personalidades da área e que nunca se concretizaram. Acreditamos que previsões na área de computação são muito difíceis de serem feitas. Optamos por criar sumários no início de cada capítulo para facilitar o processo de busca de informações.

O terminal ou **shell** nada mais é do que um aplicativo que encaminha os comandos para o Sistema Operacional. Ele interpreta os comandos enviados e retorna os resultados. Apesar de não possuir uma interface gráfica elaborada ele possui uma infinidade de funcionalidades. O conhecimento dos comandos poderá auxiliá-lo para o aumento da produtividade, pois muitas tarefas podem ser automatizadas.

Apesar de todo cuidado, podem existir alguns erros. Encontrou algum erro no livro ou tem alguma sugestão? Favor encaminhar e-mail para [alessandro.vivas@gmail.com](mailto:alessandro.vivas@gmail.com).



# Sobre os Autores



**Alessandro Vivas Andrade** é natural de Lavras/MG e atualmente reside em Diamantina/MG. É Professor do Curso de Sistemas de Informação da Universidade Federal dos Vales do Jequitinhonha e Mucuri (UFVJM) onde leciona as disciplinas de Sistemas Operacionais, Redes de Computadores e Sistemas Distribuídos. Também atua como Professor do Curso de Mestrado em Educação (PPGED/UFVJM). Graduou-se em Engenharia Elétrica (UFMG) e depois cursou Mestrado e Doutorado em Engenharia Elétrica na UFMG. Tem interesses nas áreas de Otimização, Inteligência Artificial e Redes de Computadores.



**Luciana Pereira de Assis** é natural de Belo Horizonte/MG e atualmente reside em Diamantina/MG. É Professora do Curso de Sistemas de Informação da Universidade Federal dos Vales do Jequitinhonha e Mucuri (UFVJM) onde leciona as disciplinas de Algoritmos e Estrutura de Dados, Pesquisa Operacional e Inteligência Artificial. Também atua como Professora do Curso de Mestrado em Educação (PPGED/UFVJM). Graduou-se em Ciência da Computação (UNIBH) e depois cursou Mestrado em Ciência da Computação (UFMG) e Doutorado em Engenharia Elétrica na UFMG. Tem interesses nas áreas de Otimização, Inteligência Artificial e Análise de Redes Sociais.



**André L. Maravilha** é natural de Teófilo Otoni/MG e atualmente reside em Belo Horizonte/MG. Obteve o grau de bacharel em Sistemas de Informação pela Universidade Federal dos Vales do Jequitinhonha e Mucuri (UFVJM) e o grau de mestre em Engenharia Elétrica pela Universidade Federal de Minas Gerais. Atualmente está cursando Doutorado em Engenharia Elétrica na Universidade Federal de Minas Gerais. Tem interesses nas áreas de Pesquisa Operacional, Otimização Combinatória, Heurísticas e Meta-heurísticas, e Inteligência Computacional.



# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Terminal do Mac . . . . .	2
1.1.1	Acessando o Terminal . . . . .	2
1.2	Significado do Shell . . . . .	2
1.3	Formato dos comandos . . . . .	3
1.4	Executando Comandos em Modo Privilegiado ou Superusuário . . . . .	4
1.5	Shells . . . . .	4
1.5.1	Qual é o Shell que você está usando? . . . . .	4
1.6	Alterando o Shell . . . . .	5
1.7	Movimentação no terminal . . . . .	5
1.8	Primeiros comandos . . . . .	5
1.9	Exibindo Mensagens . . . . .	6
1.10	Alterando o Nome da Máquina Temporariamente . . . . .	8
1.11	Alterando o Nome da Máquina Permanentemente . . . . .	8
1.12	Histórico do Terminal . . . . .	9
1.12.1	Comando <i>history</i> . . . . .	9
<b>2</b>	<b>Desligando o Mac OS X</b>	<b>11</b>
2.1	Saindo do sistema . . . . .	12
2.1.1	Saindo do Sistema com Logout . . . . .	12
2.1.2	Saindo do Sistema com Exit . . . . .	12
2.2	Desligando e Reiniciando o Sistema . . . . .	12
2.2.1	Desligando Imediatamente . . . . .	12
2.2.2	Desligando após um determinado tempo . . . . .	13
2.2.3	Desligando em uma hora específica . . . . .	13
2.2.4	Cancelando um shutdown . . . . .	13
2.3	Reiniciando a máquina . . . . .	14
<b>3</b>	<b>Operações em Diretórios e Arquivos</b>	<b>15</b>
3.1	Listando Arquivos e Diretórios . . . . .	16
3.1.1	Lista Simples . . . . .	16
3.1.2	Listando uma única entrada por linha . . . . .	16
3.1.3	Listando o Conteúdo no Formato Longo . . . . .	16
3.1.4	Informações sobre os arquivos e diretórios . . . . .	17
3.1.5	Obtendo informações sobre diretórios . . . . .	17
3.1.6	Listando Arquivos Ocultos . . . . .	18
3.1.7	Classificando Arquivos e Diretórios . . . . .	18
3.1.8	Imprimindo Informações sobre o Tamanho dos arquivos . . . . .	18
3.1.9	Listando Recursivamente . . . . .	19
3.2	Navegando em Diretórios . . . . .	19

3.2.1	Diretório de Trabalho . . . . .	19
3.3	Comando pwd . . . . .	20
3.4	Copiando Arquivos . . . . .	21
3.4.1	Copiando Arquivo para Diretório . . . . .	21
3.5	Copiando Múltiplos arquivos . . . . .	21
3.6	Copiando Diretórios e Sub-diretórios . . . . .	22
3.7	Movendo Arquivos . . . . .	22
3.8	Movendo e Renomeando Arquivos . . . . .	22
3.9	Criando um Arquivo Vazio com touch . . . . .	23
3.10	Apagando Arquivos . . . . .	23
3.10.1	Apagando Múltiplos Arquivos . . . . .	23
3.11	Apagando um Diretório . . . . .	23
3.12	Apagando Diretório com rmdir . . . . .	23
3.13	Criando Diretório . . . . .	24
3.14	Trabalhando com Múltiplos Diretórios . . . . .	24
3.14.1	Criando Múltiplos Diretórios . . . . .	24
3.14.2	Criar Hierarquia de Diretórios . . . . .	24
<b>4</b>	<b>Comandos para Manipulação Texto</b> . . . . .	<b>25</b>
4.1	Comando apropos . . . . .	26
4.2	Comando cat . . . . .	26
4.3	Comparando Arquivos . . . . .	27
4.3.1	Comando cmp . . . . .	27
4.3.2	Comando comm . . . . .	27
4.4	Comando csplit . . . . .	28
4.5	Cortando e Colando . . . . .	29
4.5.1	Comando cut . . . . .	29
4.5.2	Comando paste . . . . .	29
4.6	Inserindo e Retirando Tabulações . . . . .	30
4.6.1	Comando expand . . . . .	30
4.6.2	Comando unexpand . . . . .	31
4.7	Comando tr . . . . .	31
4.8	Comando fmt . . . . .	32
4.9	Comando grep . . . . .	33
4.10	Comando head . . . . .	33
4.11	Comando iconv . . . . .	33
4.12	Comando look . . . . .	34
4.13	Comando more . . . . .	34
4.14	Comando nl . . . . .	35
4.15	Comando pr . . . . .	35
4.16	Comando rev . . . . .	36
4.17	Comando stat . . . . .	36
4.18	Comando sort . . . . .	37
4.19	Comando tail . . . . .	37
4.20	Comando uniq . . . . .	38
4.21	Comando wc . . . . .	38

<b>5 Comandos de Sistema</b>	<b>41</b>
5.1 Comando alias . . . . .	42
5.2 Comando basename . . . . .	42
5.3 Shell builtin . . . . .	42
5.4 Gerando Todos os Comandos com <b>compgen</b> . . . . .	42
5.5 Comando complete . . . . .	43
5.6 Comando cron . . . . .	43
5.7 Quem sou eu e onde estou? . . . . .	44
5.7.1 Arquivo dscl . . . . .	44
5.8 Comando id . . . . .	46
5.9 Alterando a Senha . . . . .	47
5.10 Usuários Logados . . . . .	47
5.11 Comandos de Calendário, Data e Hora . . . . .	47
5.11.1 Comando Date . . . . .	47
5.11.2 Comando cal . . . . .	47
5.11.3 Comando calendar . . . . .	49
5.12 Comando chflags . . . . .	49
5.13 Comando finger . . . . .	49
5.14 Comando last . . . . .	50
5.15 Comando vm_stat . . . . .	50
5.16 Comando su . . . . .	51
5.17 Comando uname . . . . .	51
5.18 Comando uptime . . . . .	52
5.19 Variável PATH . . . . .	52
5.20 Comando w . . . . .	53
5.21 Comando whereis . . . . .	53
5.22 Comando which . . . . .	53
5.23 Procurando Informações . . . . .	53
5.23.1 Comando locate . . . . .	54
5.23.2 Comando find . . . . .	54
5.23.3 Comando whatis . . . . .	54
5.24 Rodando Comandos . . . . .	54
5.24.1 Rodando múltiplos comandos . . . . .	55
5.24.2 Rodando um comando em background . . . . .	55
<b>6 Gerenciamento de Processos</b>	<b>57</b>
6.1 Rodando Comandos em Background com <b>bg</b> e em Foreground <b>fg</b> . . . . .	58
6.2 Visualizando Todos os Processos em Execução . . . . .	59
6.3 Todos os Processos de um Usuário Específico . . . . .	59
6.4 Lista de Processos Ordenadas pelo Consumo de CPU . . . . .	60
6.5 Lista dos Processos que Mais Consomem Memória . . . . .	60
6.6 Obtendo Informações de um Processo Específico . . . . .	61
6.7 Comando top . . . . .	61
6.8 Listando todos os Sinais com o Comando kill . . . . .	62
6.9 Matando um Processo com o Comando Kill . . . . .	62
6.10 Comando killall . . . . .	62
6.11 Comando time . . . . .	63

<b>7 Permissão e Propriedade</b>	<b>65</b>
7.1 Comando chown . . . . .	66
7.2 Comando chgrp . . . . .	66
7.3 Permissão e Propriedade . . . . .	66
7.4 Comando chroot . . . . .	68
<b>8 Gerenciando Usuários</b>	<b>69</b>
8.1 Listando Todos os Usuários do Sistema . . . . .	70
8.2 Listando Grupos . . . . .	70
8.3 Adicionando Usuários . . . . .	70
8.4 Definindo Senha para Novos Usuários . . . . .	71
8.5 Apagando uma Conta de Usuário . . . . .	71
8.6 Adicionando um Novo Grupo . . . . .	72
8.7 Deletando um Grupo . . . . .	72
<b>9 Comandos para Redes de Computadores</b>	<b>73</b>
9.1 Comando hostname . . . . .	74
9.2 Comando curl . . . . .	74
9.3 Comando e Tabela ARP . . . . .	74
9.4 Listando Todas as Interfaces de Rede . . . . .	75
9.5 Trabalhando com o Endereço IP . . . . .	75
9.5.1 Verificando Endereço IP . . . . .	75
9.5.2 Habilitando e Desabilitando a Interface de Rede . . . . .	77
9.5.3 Alterando a MTU de uma Interface . . . . .	78
9.5.4 Alterando Endereço IP . . . . .	78
9.5.5 Descobrir endereço IP de um Determinado Host . . . . .	79
9.6 Comando ping . . . . .	79
9.7 Informações sobre Domínios . . . . .	80
9.7.1 Comando dig . . . . .	80
9.7.2 Comando nslookup . . . . .	81
9.8 Traçando caminhos de um host a outro . . . . .	81
9.8.1 Descobrindo o Endereço do seu Roteador sem Fio . . . . .	82
9.9 Comando netstat . . . . .	82
9.9.1 Tabela de Roteamento . . . . .	92
9.10 Comando nettop . . . . .	93
9.11 Network Mapper . . . . .	93
9.11.1 Instalação . . . . .	93
9.11.2 Analisando portas abertas . . . . .	93
9.11.3 Comando nmap com opção de mais informações . . . . .	94
9.11.4 Rastreando Múltiplos Hosts . . . . .	94
9.12 Comando route . . . . .	96
9.13 Comando telnet . . . . .	97
9.13.1 Acessando Servidor Web via Telnet . . . . .	97
9.14 Acesso Remoto com ssh . . . . .	101
9.14.1 Acesso Remoto . . . . .	101
9.15 Copiando Arquivos com scp . . . . .	101
9.16 Copiando um Diretório de um Servidor Remoto . . . . .	101
9.17 Comando tcpdump . . . . .	102
9.17.1 Listando Todas as Interfaces de Rede . . . . .	102
9.17.2 Realizando Captura de Pacotes da Interface Sem Fio . . . . .	102

9.17.3	Capturando Pacotes de um Host Específico . . . . .	103
9.17.4	Capturando Pacotes de uma Rede . . . . .	103
9.18	Baixando Sites com wget . . . . .	105
<b>10</b>	<b>Compactação e Backup</b>	<b>107</b>
10.1	Compactando e Descompactando com zip . . . . .	108
10.1.1	Compactando com zip . . . . .	108
10.1.2	Descompactando com zip . . . . .	108
10.1.3	Informações sobre a compactação . . . . .	108
10.2	Compactando e Descompactando com .tar . . . . .	109
10.2.1	Compactando com tar (gerando arquivos .tar) . . . . .	109
10.2.2	Descompactando com tar (arquivos .tar) . . . . .	109
10.3	Compactando e Descompactando com .tar.gz . . . . .	109
10.3.1	Compactando com tar e gzip (gerando arquivos .tar.gz) . . . . .	109
10.3.2	Descompactando com tar e gzip (arquivos .tar.gz) . . . . .	109
10.4	Compactando e Descompactando com .tar.bz2 . . . . .	109
10.4.1	Compactando com tar e gzip (gerando arquivos .tar.bz2) . . . . .	109
10.4.2	Descompactando com tar e gzip (arquivos .tar.bz2) . . . . .	110
10.5	Manipulando Imagens .dmg . . . . .	110
10.5.1	Criando Imagens . . . . .	110
10.5.2	Duplicando uma Imagem . . . . .	111
<b>11</b>	<b>Comandos Úteis</b>	<b>113</b>
11.1	Comando say . . . . .	114
11.1.1	Passando uma Frase . . . . .	114
11.1.2	Passando um Arquivo como Argumento . . . . .	114
11.1.3	Passando o Locutor como Argumento . . . . .	114
11.2	Imprimir Metadados com mtlst . . . . .	114
11.3	Comando mdfind . . . . .	115
11.4	Comando units . . . . .	116
11.5	Comando yes . . . . .	117
11.6	Linguagem awk . . . . .	117
<b>12</b>	<b>Comandos para Sistema de Arquivos</b>	<b>121</b>
12.1	Tamanho de Arquivos e Diretórios . . . . .	122
12.2	Espaço Disponível no Disco . . . . .	122
12.3	Criando Atalhos . . . . .	123
12.4	Entendendo Discos e Partições . . . . .	123
12.5	Imprimir Tabela de Partições do Linux . . . . .	123
12.6	Obtendo Informações sobre o Disco com fdisk . . . . .	124
12.7	Comando sfdisk . . . . .	124
12.8	Listando Informações sobre as Partições com lsblk . . . . .	124
12.9	Listando Informações sobre Permissões do Disco com lsblk . . . . .	124
<b>13</b>	<b>Comandos para Analisar o Desempenho</b>	<b>127</b>
13.1	Analizando Consumo de CPU com o Comando sar . . . . .	128
13.2	Analizando de Desempenho com iostat . . . . .	128
13.3	Informações de Memória de Processos em Execução . . . . .	128
13.4	Analizando a Memória com vm_stat . . . . .	130
13.5	Comando top . . . . .	130
13.6	Comando powermetrics . . . . .	131

<b>14 Configuração de Hardware e Software</b>	<b>133</b>
14.1 Visualizando Informações de Boot . . . . .	134
14.2 Visualizando Informações sobre o Sistema . . . . .	134
14.3 Visualizando Informações sobre a Versão do Kernel . . . . .	134
14.4 Informações sobre a Versão do Software . . . . .	135
14.5 Informações de Hardware com hostinfo . . . . .	135
14.6 Informações sobre o Sistema . . . . .	135
14.7 Visualizando Informações sobre a sua CPU . . . . .	136
14.8 Visualizando todas as Variáveis do Kernel . . . . .	137
14.9 Visualizando Informações sobre os Dispositivos USB . . . . .	138
14.10 Listando Todos os Dispositivos PCI . . . . .	140
14.11 Verificando Todas as Partições . . . . .	140
14.12 Obtendo Informações sobre a Memória . . . . .	141
14.13 Listando Impressoras do Sistema . . . . .	141

# Listas de Figuras

1.1	Busca do Terminal no <i>MAC OS X</i>	2
1.2	Janela do Terminal do <i>MAC OS X</i>	2
5.1	Calendário de 2030	48
9.1	Comando nettop	93
12.1	Comando fdisk	125
12.2	Comando sfdisk	125
12.3	Comando lsblk	125
12.4	Listando Permissões com lsblk	126
13.1	Uso do top para Obter Estatísticas de CPU	130



# **Lista de Tabelas**

1.1	Atalhos para o Terminal . . . . .	6
1.2	Atalhos para o Terminal . . . . .	7



# Listas de Listagens

1.1	Prompt de Login . . . . .	2
1.2	Explicando a Padronização de Apresentação . . . . .	3
1.3	Formato dos Comandos . . . . .	3
1.4	Opções dos Comandos . . . . .	3
1.5	Opções Múltiplas . . . . .	4
1.6	Shell Utilizado . . . . .	4
1.7	<i>Shells</i> Instalados . . . . .	4
1.8	Alterando para o Shell <b>tsh</b> . . . . .	5
1.9	Shell <b>tsh</b> . . . . .	5
1.10	Alterando para o Shell <b>tsh</b> . . . . .	5
1.11	Caminho Completo . . . . .	6
1.12	Limpando a Tela . . . . .	6
1.13	Exibindo Mensagens com echo . . . . .	6
1.14	Comando echo com aspas simples . . . . .	6
1.15	Comando echo com aspas duplas . . . . .	7
1.16	Visualizando Variáveis de Ambiente . . . . .	7
1.17	Saltando Linhas com echo . . . . .	7
1.18	Tabulação Horizontal . . . . .	7
1.19	Tabulação Vertical . . . . .	7
1.20	Barra Invertida . . . . .	7
1.21	Retorno de Linha . . . . .	8
1.22	Alterando o Nome da Máquina . . . . .	8
1.23	Visualizando o Nome da Máquina . . . . .	8
1.24	Resultado da Alteração do Nome da Máquina . . . . .	8
1.25	Alteração Permanentemente . . . . .	8
1.26	Verificando a Alteração do Nome . . . . .	8
1.27	Histórico . . . . .	9
1.28	Comando history . . . . .	9
1.29	Executar um determinado comando do history . . . . .	9
1.30	Exemplo de busca no history . . . . .	10
1.31	Limpar o histórico . . . . .	10
1.32	Tamanho do histórico . . . . .	10
2.1	Comando logout . . . . .	12
2.2	Comando exit . . . . .	12
2.3	Desligando Imediatamente com shutdown . . . . .	13
2.4	Desligando Após Determinado Intervalo de Tempo . . . . .	13
2.5	Mensagens recebidas . . . . .	13
2.6	Desligando Imediatamente . . . . .	13
2.7	Desligando em 5 minutos . . . . .	13
2.8	Cancelando Shutdown . . . . .	13
2.9	Cancelando Shutdown . . . . .	14

2.10	Reiniciando com reboot . . . . .	14
2.11	Reiniciando com shutdown . . . . .	14
2.12	Reiniciando após determinado tempo . . . . .	14
2.13	Reiniciando em uma hora específica . . . . .	14
3.1	Listando o Conteúdo de um Diretório . . . . .	16
3.2	Uma Entrada por Linha . . . . .	16
3.3	Comando ls no formato longo . . . . .	16
3.4	Símbolo - . . . . .	17
3.5	Símbolo d . . . . .	17
3.6	Símbolo l . . . . .	17
3.7	Obtendo informações sobre diretórios . . . . .	17
3.8	Listando Todos os Arquivos Inclusive os Ocultos . . . . .	18
3.9	Classificando Arquivos e Diretórios . . . . .	18
3.10	Tamanho em Blocos . . . . .	18
3.11	Tamanho dos Arquivos . . . . .	18
3.12	Combinando Opções do Comando ls . . . . .	19
3.13	Visualizando Informações sobre o Diretório . . . . .	19
3.14	Listando Recursivamente . . . . .	19
3.15	Comando cd . . . . .	20
3.16	Atalho para o Diretório Raiz do Usuário . . . . .	20
3.17	Significado de ~ . . . . .	20
3.18	Comando cd Sem Opções . . . . .	20
3.19	Retornando ao Diretório do Usuário . . . . .	20
3.20	Comando pwd . . . . .	20
3.21	Outro Exemplo do Comando pwd . . . . .	21
3.22	Sintaxe do Comando cp . . . . .	21
3.23	Copiando Arquivo para Diretório . . . . .	21
3.24	Copiando Arquivo para um Diretório . . . . .	21
3.25	Copiando Múltiplos Arquivos . . . . .	22
3.26	Copiando um Arquivo em Outro . . . . .	22
3.27	Copiando com a Opção -i . . . . .	22
3.28	Copiando Diretórios . . . . .	22
3.29	Movendo Arquivos com mv . . . . .	22
3.30	Renomeando Arquivos com rename . . . . .	22
3.31	Criando Arquivos com touch . . . . .	23
3.32	Apagando um Arquivo . . . . .	23
3.33	Apagando Múltiplos Arquivos . . . . .	23
3.34	Apagando Diretório . . . . .	23
3.35	Apagando Diretório com rmdir . . . . .	24
3.36	Criando Diretório com mkdir . . . . .	24
3.37	Criando Múltiplos Diretórios . . . . .	24
3.38	Criando Árvore de Diretórios . . . . .	24
4.1	Comando apropos . . . . .	26
4.2	Comando cat . . . . .	26
4.3	Comando cat . . . . .	26
4.4	Arquivos cidades1.txt e cidades.txt . . . . .	27
4.5	Arquivos cidades1.txt e cidades.txt . . . . .	27
4.6	Comando comm . . . . .	27
4.7	Comando csplit . . . . .	28
4.8	Comando csplit . . . . .	28

4.9	Comando cut . . . . .	29
4.10	Arquivo com Cidades . . . . .	29
4.11	Separando dados de um Arquivo . . . . .	29
4.12	Arquivos de Exemplo para o Comando paste . . . . .	29
4.13	Combinando Dois Arquivos com o Comando paste . . . . .	30
4.14	Listagem Original . . . . .	30
4.15	Tabulações Convertidas para 1 Espaço . . . . .	30
4.16	Convertendo Tabulações . . . . .	30
4.17	Listagem Original . . . . .	31
4.18	Convertendo Espaços em Tabulações . . . . .	31
4.19	Convertendo tabulações em espaço simples . . . . .	31
4.20	Convertendo tabulações em espaço simples e removendo múltiplas ocorrências de espaços . . . . .	31
4.21	Convertendo maiúsculas em minúsculas . . . . .	31
4.22	Outra forma de converter MAIÚSCULA em minúsculas . . . . .	32
4.23	Transformar espaços em quebra de linha . . . . .	32
4.24	Substituir chaves por parênteses . . . . .	32
4.25	Duas maneiras para se remover dígitos . . . . .	32
4.26	Formatando Linhas com o Comando fmt . . . . .	32
4.27	Listagem para uso do grep . . . . .	33
4.28	Imprime as Linhas Iniciais de um Arquivo . . . . .	33
4.29	Imprime as Duas Linhas Iniciais de um Arquivo . . . . .	33
4.30	Convertendo Padrões de Caracteres . . . . .	34
4.31	Comando look . . . . .	34
4.32	Comando look . . . . .	34
4.33	Visualizando Arquivos Longos . . . . .	35
4.34	Contando o Número de Linhas . . . . .	35
4.35	Dividindo em Colunas . . . . .	36
4.36	Comando para Inverter os Caracteres - rev . . . . .	36
4.37	Invertendo Caracteres com rev . . . . .	36
4.38	Comando stat . . . . .	37
4.39	Ordenando Arquivos com sort . . . . .	37
4.40	Ordenando Arquivos com sort . . . . .	37
4.41	Imprime as Linhas Finais de um Arquivo . . . . .	37
4.42	Imprime as Duas Linhas Finais de um Arquivo . . . . .	38
4.43	Comando uniq . . . . .	38
4.44	Contando quantas palavras distintas existem em um texto . . . . .	38
4.45	Contando o Número de Caracteres . . . . .	39
4.46	Contando o Número de Linhas . . . . .	39
4.47	Contando o Número de Palavras . . . . .	39
4.48	Contando o Número de Bytes . . . . .	39
5.1	Criando Apelidos para Comandos . . . . .	42
5.2	Comando basename . . . . .	42
5.3	Comando builtin . . . . .	42
5.4	Comando whoami . . . . .	42
5.5	Opções Múltiplas . . . . .	43
5.6	Agendamento com crontab . . . . .	44
5.7	Listando os Agendamentos . . . . .	44
5.8	Comando whoami e pwd . . . . .	44
5.9	Listando Usuários . . . . .	44
5.10	Informações Detalhadas . . . . .	46

5.11	Identificadores no Mac . . . . .	46
5.12	Alterando a Senha . . . . .	47
5.13	Usuários Logados . . . . .	47
5.14	Visualizando Data e hora . . . . .	47
5.15	Comando cal . . . . .	47
5.16	Imprimindo o Ano de 2030 . . . . .	48
5.17	Imprime Mês e Ano . . . . .	48
5.18	Domingo como Primeiro Dia da Semana . . . . .	48
5.19	Atributos do Arquivo cidades.txt . . . . .	49
5.20	Alterando o Estado do Arquivo . . . . .	49
5.21	Atributos do Arquivo cidades.txt . . . . .	49
5.22	Alterando os Atributos do Arquivo para Visível . . . . .	49
5.23	Comando finger . . . . .	49
5.24	Comando finger -l . . . . .	50
5.25	Comando finger no Linux . . . . .	50
5.26	Comando last . . . . .	50
5.27	Comando free . . . . .	50
5.28	Logar como Super Usuário . . . . .	51
5.29	Verificar Informações sobre o Mac . . . . .	51
5.30	Verificar sua Versão do kernel . . . . .	51
5.31	Verificando a Plataforma . . . . .	51
5.32	Verificar o Nome de sua Máquina . . . . .	52
5.33	Apresenta todas as informações sobre seu sistema operacional . . . . .	52
5.34	Tempo de Funcionamento . . . . .	52
5.35	Localização de um Comando . . . . .	52
5.36	Variável PATH . . . . .	52
5.37	Comando w . . . . .	53
5.38	Comando whereis . . . . .	53
5.39	Comando which . . . . .	53
5.40	Comando locate . . . . .	54
5.41	Comando find . . . . .	54
5.42	Comando whatis . . . . .	54
5.43	Rodando Múltiplos Comandos . . . . .	55
5.44	Rodando Múltiplos Comandos Condicionados . . . . .	55
5.45	Rodando Comando em Background . . . . .	55
6.1	Rodar Comandos em Background . . . . .	58
6.2	Rodando em Foreground . . . . .	58
6.3	Uso do <b>bg</b> . . . . .	58
6.4	Descubra o PID . . . . .	58
6.5	Pare a Execução do Processo . . . . .	58
6.6	Parando a Execução do Comando . . . . .	59
6.7	Executando novamente o processo . . . . .	59
6.8	Todos os Processos em Execução . . . . .	59
6.9	Todos os Processos em Execução de um Usuário Específico . . . . .	60
6.10	Processos que Mais Consomem CPU . . . . .	60
6.11	Processos que Mais Consomem CPU . . . . .	61
6.12	Obtendo Informações de um Processo Específico . . . . .	61
6.13	Comando top . . . . .	61
6.14	Comando kill . . . . .	62
6.15	Comando kill . . . . .	62

6.16	Matando o Processo Firefox . . . . .	62
6.17	Utilizando killall . . . . .	63
6.18	Calculando Tempo de Execução de um Programa ou Script . . . . .	63
6.19	Calculando Tempo de Execução de um Programa ou Script . . . . .	63
7.1	Listando o Proprietário e as Permissões dos Arquivos . . . . .	66
7.2	Alterando o Proprietário do Arquivo . . . . .	66
7.3	Verificando o Grupo . . . . .	66
7.4	Alterando o Grupo do Arquivo . . . . .	66
7.5	Listando o Proprietário e as Permissões dos Arquivos . . . . .	66
7.6	Trocando o Dono de um Arquivo . . . . .	67
7.7	Trocando as Permissões de um Arquivo . . . . .	67
7.8	Exemplos de Utilização do chmod . . . . .	67
8.1	Listando Todos os Usuários do Linux . . . . .	70
8.2	Listando Todos os Usuários do Linux . . . . .	70
8.3	Adicionando Usuários . . . . .	71
8.4	Definindo Informações Adicionais . . . . .	71
8.5	Definindo Senha de Usuário . . . . .	71
8.6	Apagando Usuários . . . . .	71
8.7	Adicionando um Novo Grupo . . . . .	72
8.8	Comandos para Manipulação de Grupos . . . . .	72
8.9	Apagando um Grupo . . . . .	72
9.1	Comando hostname . . . . .	74
9.2	Download de uma Página . . . . .	74
9.3	Tabela ARP . . . . .	74
9.4	Lista todas as Interfaces . . . . .	75
9.5	Verificando o Endereço IP . . . . .	75
9.6	Verificando o Endereço de uma Interface Específica . . . . .	76
9.7	Verificando Status da Rede Ethernet . . . . .	77
9.8	Desabilitando a Interface de Rede Ethernet . . . . .	77
9.9	Verificando a Ação Realizada na Listagem 9.8 . . . . .	77
9.10	Habilitando a Interface de Rede Ethernet . . . . .	77
9.11	Verificando a Ação Realizada na Listagem 9.10 . . . . .	77
9.12	Verificando MTU da Rede Ethernet . . . . .	78
9.13	Alterando o MTU da Placa de Rede . . . . .	78
9.14	Verificando o Endereço IP . . . . .	78
9.15	Alterando Endereço IP . . . . .	78
9.16	Descobrindo o Endereço Ip de um Host . . . . .	79
9.17	Comando ping . . . . .	79
9.18	Exemplo do Comando ping . . . . .	79
9.19	Comando ping . . . . .	80
9.20	Comando ping com Opção de Tempo . . . . .	80
9.21	Descobrir Informações sobre um Domínio . . . . .	80
9.22	Utilizando o comando nslookup . . . . .	81
9.23	Rotas . . . . .	81
9.24	Endereço do Roteador sem Fio . . . . .	82
9.25	Estatísticas de Rede com netstat . . . . .	82
9.26	Comando netstat -i . . . . .	91
9.27	Visualizando Tabela de Roteamento com netstat . . . . .	92
9.28	Verificando Portas Abertas . . . . .	93
9.29	Comando nmap com opção -v . . . . .	94

9.30 Rastreando Múltiplos Hosts. <i>numbers</i>	94
9.31 Rastreando Múltiplos Domínios	95
9.32 Rastreando uma Sub-rede	95
9.33 Adicionando uma rota	96
9.34 Visualizando a Tabela de Roteamento	96
9.35 Visualizando a Tabela de Roteamento	96
9.36 Visualizando a Tabela de Roteamento	96
9.37 Uso do Telnet	97
9.38 Resposta do Servidor	97
9.39 Acessando com Telnet o Servidor Web	97
9.40 Resposta do Servidor	97
9.41 Utilizando o ssh	101
9.42 Copiando Arquivo em Servidor Remoto	101
9.43 Copiando um Diretório de um Servidor Remoto	102
9.44 Interfaces que podem ser utilizadas com <b>tcpdump</b>	102
9.45 Capturando Pacotes da Interface de Rede Sem Fio	102
9.46 Capturando Pacotes da Interface de um Host Específico	103
9.47 Capturando Pacotes da Interface de um Host Específico	103
9.48 Baixando Sites com <b>wget</b>	105
10.1 Compactando um Diretório com <b>zip</b>	108
10.2 Compactando arquivos com <b>zip</b>	108
10.3 Descompactando um Diretório com <b>unzip</b>	108
10.4 Verificando Informações com <b>zipinfo</b>	108
10.5 Compactando um Diretório com <b>tar</b>	109
10.6 Descompactando um arquivo com extensão <b>tar</b>	109
10.7 Compactando um Diretório com <b>tar.gz</b>	109
10.8 Descompactando um arquivo com extensão <b>tar.gz</b>	109
10.9 Compactando um Diretório com <b>tar.gz</b>	109
10.10 Descompactando um arquivo com extensão <b>tar.bz2</b>	110
10.11 Compactando com <b>bzip2</b>	110
10.12 Descompactando com <b>bzip2</b>	110
10.13 Criando Imagem <b>dmg</b>	110
10.14 Criando Imagem <b>dmg</b>	111
10.15 Ejetando Volume <b>dmg</b>	111
10.16 Copiando Imagem com <b>asr</b>	111
11.1 Frase específica com o comando <b>say</b>	114
11.2 Arquivo com o comando <b>say</b>	114
11.3 Selecionando locutor com o comando <b>say</b>	114
11.4 Lendo Metadados com <b>mtls</b>	114
11.5 Exemplos de utilização do comando <b>mdfind</b>	116
11.6 Exemplos de utilização do comando <b>mdfind</b>	116
11.7 Exemplos de utilização do comando <b>units</b>	116
11.8 Exemplo de utilização do comando <b>yes</b> para responder automaticamente a perguntas com ‘yes’	117
11.9 Imprimindo uma mensagem indefinidamente no terminal utilizando o <b>yes</b>	117
11.10 Utilizando o Comando <b>yes</b> para Processamento de Latex	117
11.11 Imprimindo a Coluna 1	117
11.12 Imprimindo a Coluna 1	117
11.13 Imprimindo a Coluna 1	118
11.14 Contando o Número de Ocorrências por Cidade	118
12.1 Tamanho de Arquivos em Blocos	122

12.2 Tamanho de Arquivos . . . . .	122
12.3 Espaço Disponível no Disco . . . . .	122
12.4 Espaço Disponível no Disco em Blocos . . . . .	122
12.5 Criando um Link . . . . .	123
12.6 Listando o Link . . . . .	123
12.7 Criando e Listando um Hard Link . . . . .	123
12.8 Listando as Partições do Disco Rígido . . . . .	124
13.1 Analisando Desempenho da CPU com sar . . . . .	128
13.2 Analisando Desempenho de Todos os Núcleos com mpstat . . . . .	128
13.3 Analisando Memória com vmmap . . . . .	129
13.4 Analisando a Memória com vm_stat . . . . .	130
13.5 Analisando Informações sobre Energia . . . . .	131
14.1 Informações sobre boot . . . . .	134
14.2 Disco de boot . . . . .	134
14.3 Resultado do Comando dmesg . . . . .	134
14.4 Versão do Kernel . . . . .	135
14.5 Arquitetura do Hardware . . . . .	135
14.6 Informações do Software . . . . .	135
14.7 Informações Gerais sobre seu Sistema . . . . .	135
14.8 Informações do Sistema . . . . .	136
14.9 Informações do CPU . . . . .	136
14.10 Informações das Variáveis do Kernel . . . . .	137
14.11 Imprimindo uma Variável Específica . . . . .	137
14.12 Lista Dispositivos USB . . . . .	138
14.13 Listando USB . . . . .	138
14.14 Listando PCI . . . . .	140
14.15 Listando PCI . . . . .	140
14.16 Imprimindo as Partições . . . . .	141
14.17 Listando Informações sobre a Memória . . . . .	141
14.18 Listando Informações sobre Impressoras . . . . .	141
14.19 Listando a Impressora Padrão . . . . .	141



# Introdução

*A Apple é uma bagunça caótica  
sem visão estratégica e  
certamente não tem futuro.*

Revista TIME, 1996

## Sumário

1.1	Terminal do Mac . . . . .	2
1.2	Significado do Shell . . . . .	2
1.3	Formato dos comandos . . . . .	3
1.4	Executando Comandos em Modo Privilegiado ou Superusuário . . . . .	4
1.5	Shells . . . . .	4
1.6	Alterando o Shell . . . . .	5
1.7	Movimentação no terminal . . . . .	5
1.8	Primeiros comandos . . . . .	5
1.9	Exibindo Mensagens . . . . .	6
1.10	Alterando o Nome da Máquina Temporariamente . . . . .	8
1.11	Alterando o Nome da Máquina Permanentemente . . . . .	8
1.12	Histórico do Terminal . . . . .	9

## 1.1 Terminal do Mac

O terminal ou *Shell* nada mais é do que um aplicativo que encaminha os comandos para o Sistema Operacional. Ele interpreta dos comandos enviados e retorna os resultados. Apesar de não possuir uma interface gráfica elaborada ele possui uma infinidade de funcionalidades. O conhecimento dos comandos poderá auxiliá-lo para o aumento da produtividade, pois muitas tarefas podem ser automatizadas.

### 1.1.1 Acessando o Terminal

O terminal do *MAC OS X* pode ser acessado através da interface gráfica como na Figura 1.1.

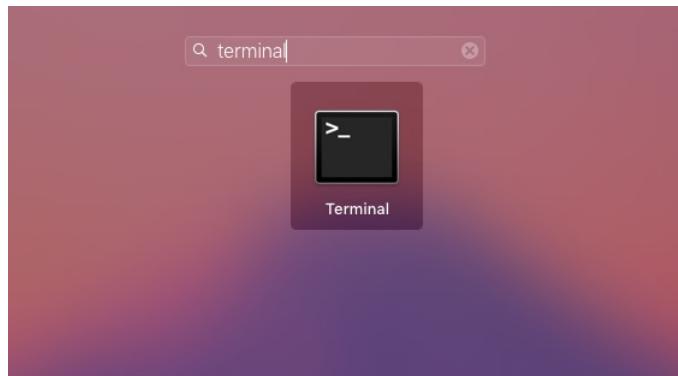


Figura 1.1: Busca do Terminal no *MAC OS X*

Clique no ícone do terminal e abrirá a janela como na Figura 1.2.

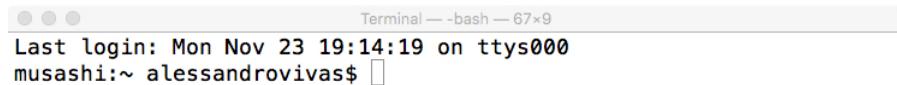


Figura 1.2: Janela do Terminal do *MAC OS X*

Pronto! A partir de agora é só digitar os comandos. O *MAC OS X* não possui terminais virtuais e por isto todo acesso deve ser feito através da interface gráfica. O Linux permite que o usuário acesse diversos terminais virtuais através dos comandos <CONTROL+ALT+Fn>, onde Fn pode variar de F1 até F5. Para voltar para a interface gráfica digite <CONTROL+ALT+F6>.

## 1.2 Significado do Shell

Ao abrir o terminal irá aparecer algumas informações, como na Listagem 1.1.

### Listagem 1.1: Prompt de Login

```
Last login: Mon Nov 23 19:26:26 on ttys000
musashi:~ alessandrovivas$
```

Na Figura acima são exibidas informações da última vez que o usuário se autenticou no sistema. Após isso, é encontrado o nome da máquina (musashi), um sinal de dois pontos, o diretório corrente (, que significa que o diretório corrente é o diretório do usuário), e o nome do usuário (alessandrovivas). Em seguida é encontrado o sinal \$, que indica que o terminal está pronto para receber comandos digitados pelo usuário (prompt).

#### Listagem 1.2: Explicando a Padronização de Apresentação

```
/Users/alessandrovivas$
```

Diretórios no *MAC OS X* e *Linux/UNIX* são especificados por uma / e não uma \, diferentemente do que é definido para outros sistemas, como por exemplo o DOS.

## 1.3 Formato dos comandos

Os comandos (arquivos executáveis, chamados também de programas) no *MAC OS X*, passados via *Shell*, possuem a seguinte forma:

- <comando> → ls
- <comando><espaço><opções> → ls -lF
- <comando><espaço><opções><espaço><argumentos> → cp -R /home/vivas/teste /home/vivas/ensino/

A quase totalidade dos comandos possui todos os três elementos acima. A Listagem 1.3 apresenta o comando de copiar um arquivo de um diretório para outro arquivo que está em outro diretório, preservando os atributos do arquivo (permissões, dono, marca de tempo).

#### Listagem 1.3: Formato dos Comandos

```
cp -p /Users/origem/texto.txt /Users/destino/texto.txt
```

Entretanto, existem alguns comandos que possuem apenas opções, apenas argumentos ou nenhum destes, i.e., somente o próprio comando é necessário. O comando clear , utilizado para limpar a tela do terminal, não possui argumento algum.

É importante atentar para a existência ou não de espaços entre os caracteres ao se definir uma ação completa (comando <espaço> opção1 <espaço> opção2 <espaço> opção3 <espaço> argumento1 <espaço> argumento2 <espaço> argumento3).

Existe uma flexibilidade para se passar opções ao sistema operacional. Quando há a necessidade de se especificar mais de uma opção o usuário pode utilizar um aninhamento de opções, i.e., usar o mesmo hífen para especificar mais de uma opção:

- <comando> -<opção1> -<opção2> -<opção3>
- <comando> -<opção1><opção2><opção3>

Toda opção é precedida de um ou dois hífens(-), colocado sem espaços. Os dois hífens são usados para opções por extenso como apresentado na Listagem 1.4.

#### Listagem 1.4: Opções dos Comandos

```
$ls<espaço>-a<espaço>-l  
$ls<espaço>--all  
$pwd<espaço>--version  
$cd<espaço>--help
```

A opção é definida alternativamente por uma letra (-a, -l,-H) ou por extenso(–color,–size,–count). O uso de dois hifens isolados na linha de comando estabelece para o interpretador que não haverá mais opções a serem passadas para se efetuar aquele comando. Caso exista algo do tipo -texto escrito logo em seguida aos dois hífens em sequência, este não será interpretado pelo *Shell* como uma opção, mas sim como um argumento. A Listagem 1.5 apresenta um exemplo, no qual a opção -F não será interpretada como opção, conforme explicado anteriormente.

#### Listagem 1.5: Opções Múltiplas

```
$ls -la -- -F
```

## 1.4 Executando Comandos em Modo Privilegiado ou Superusuário

Assim como o Linux, o Mac OS X possui dois tipos de contas : usuário normal e superusuário. O superusuário tem o poder de fazer modificações no sistema operacional e fazer operações privilegiadas. Muitos comandos não são permitidos para execução e necessitam de acesso especial. Para conseguir este acesso especial você precisa digitar **sudo** antes de qualquer comando e digitar a senha do administrador da máquina.

## 1.5 Shells

Ao entrar no *Mac OS X*, o sistema habilita um *Shell* para trabalho. O *Shell* (interpretador de comandos) default, especificado pela configuração inicial, é o bash (localizado no diretório */bin*), mas pode-se alterar essa escolha.

- bash - \$ - Bourne Again Shell. O *Shell* mais utilizado do *Mac OS X*. Criado e distribuído pelo projeto GNU. Oferece comandos de edição de linha, substituição baseado no histórico e compatibilidade com o Bourne *Shell*(sh).
- csh - % C *Shell*. Desenvolvido em Berkeley. Compatível com Bourne Shell para uso interativo, mas tem uma interface diferente de programação. Não oferece comandos de edição.
- ksh - Korn *Shell* - O *Shell* mais popular do Unix e o primeiro a introduzir as técnicas modernas de *Shell* no Bourne *Shell*. Oferece comandos de edição de linha.
- sh - \$ - Bourne Shell. Shell original do Linux. Não oferece comandos de edição.
- zsh - z *Shell*. O mais novo dos *Shells*. Compatível com Bourne *Shell* e oferece comandos de edição.
- tcsh - % - um C *Shell* melhorado.

### 1.5.1 Qual é o Shell que você está usando?

Para saber qual *Shell* você está utilizando basta digitar o comando apresentado na Listagem 1.6. O comando echo é utilizado para imprimir variáveis de ambiente ou textos no terminal.

#### Listagem 1.6: Shell Utilizado

```
[musashi:~ alessandrovivas$ echo $SHELL  
/bin/bash
```

Para saber quais interpretadores de comandos estão instalados em seu sistema utilize o comando apresentado na Listagem 1.7. O comando cat é utilizado para visualizar o conteúdo de um arquivo.

#### Listagem 1.7: Shells Instalados

```
musashi-2: alessandrovivas$ cat /etc/shells
# List of acceptable shells for chpass(1).
# Ftpd will not allow users to connect who are not using
# one of these shells.

/bin/bash
/bin/csh
/bin/ksh
/bin/sh
/bin/tcsh
/bin/zsh
```

---

## 1.6 Alterando o Shell

Para mudar o *Shell* deve-se utilizar o comando `chsh`. A Listagem 1.8 ilustra como alterar o *Shell* para tcsh

### Listagem 1.8: Alterando para o Shell tcsh

```
musashi:~ alessandrovivas$ chsh -s /bin/tcsh
Changing shell for alessandrovivas.
Password for alessandrovivas:
```

As alterações só serão realizadas quando outro terminal for aberto como na Listagem 1.9

### Listagem 1.9: Shell tcsh

```
Last login: Mon Nov 23 20:57:46 on ttys001
[musashi:~] alessandrovivas% echo $SHELL
/bin/tcsh
[musashi:~] alessandrovivas%
```

Para alterar para o zsh utilize a Listagem 1.10.

### Listagem 1.10: Alterando para o Shell zsh

```
musashi:~ alessandrovivas$ chsh -s /bin/zsh
Changing shell for alessandrovivas.
Password for alessandrovivas:
```

## 1.7 Movimentação no terminal

Na Tabela 1.1 estão listados os atalhos para movimentação no terminal.

## 1.8 Primeiros comandos

Um comando é um software que realiza uma determinada função - usualmente uma função especializada. No *Mac OS X*, comando é um simples arquivo localizado geralmente no diretório `/bin` ou `/sbin`. Assim, define-se como caminho absoluto aquele caminho completo, desde o diretório raiz. A Listagem 1.11 apresenta o caminho completo para o comando `ls`.

Atalho	Descrição da funcionalidade
<b>&lt;setas direcionais&gt;</b>	Mover o cursor.
<b>&lt;control + a&gt;</b>	Mover o cursor para o início da linha de comandos
<b>&lt;control + e&gt;</b>	Mover o cursor para o fim da linha de comandos
<b>&lt;control + p&gt;</b>	Busca o comando anterior
<b>&lt;control + n&gt;</b>	Busca o próximo comando
<b>&lt;control + b&gt;</b>	Mover o cursor um caractere para a esquerda.
<b>&lt;control + f&gt;</b>	Mover o cursor um caractere para a direita.
<b>&lt;control + l&gt;</b>	Apaga a tela.
<b>&lt;control + u&gt;</b>	Apaga os caracteres a esquerda do cursor
<b>&lt;control + k&gt;</b>	Apaga os caracteres a direita do cursor
<b>&lt;control + delete&gt;</b>	Apaga o caractere abaixo do cursor.
<b>&lt;backspace&gt;</b>	Apaga caractere a esquerda.
<b>&lt;control + h&gt;</b>	Apaga caractere a esquerda.
<b>&lt;control + w&gt;</b>	Corta e copia para o <i>clipboard</i> .
<b>&lt;control + y&gt;</b>	Cola o conteúdo do <i>clipboard</i>
<b>&lt;control + c&gt;</b>	Interrompe a execução de um comando
<b>&lt;control + r&gt;</b>	Busca comando no histórico.
<b>&lt;COMAND+↑&gt;</b>	Navegação entre comandos.
<b>&lt;COMAND+↓&gt;</b>	Navegação entre comandos.

Tabela 1.1: Atalhos para o Terminal

#### Listagem 1.11: Caminho Completo

```
musashi:~ alessandrovivas$ /bin/ls
```

A seguir são dados os comandos mais básicos. Para limpar a tela do terminal, use o comando **clear** ou a combinação <CONTROL+L>. O cursor será posicionado no canto superior esquerdo:

#### Listagem 1.12: Limpando a Tela

```
musashi:~ alessandrovivas$ clear
```

## 1.9 Exibindo Mensagens

O comando **echo** disponibiliza mensagens na saída padrão (vídeo). O comando não precisa de opções e como argumento recebe o texto como na Listagem 1.13.

#### Listagem 1.13: Exibindo Mensagens com echo

```
andarilho: alessandrovivas$ echo alessandro vivas andrade
alessandro vivas andrade
```

Pode ser acompanhado de aspas simples, Listagem 1.14.

#### Listagem 1.14: Comando echo com aspas simples

```
andarilho: alessandrovivas$ echo
alessandro vivas andrade
```

Pode ser acompanhado de aspas duplas, Listagem 1.15.

#### Listagem 1.15: Comando echo com aspas duplas

```
andarilho: alessandrovivas$ echo  
alessandro vivas andrade
```

Pode ser utilizado para visualização de variáveis de ambiente como na Listagem 1.16.

#### Listagem 1.16: Visualizando Variáveis de Ambiente

```
andarilho: alessandrovivas$ echo $PATH  
/Library/Frameworks/Python.framework/Versions/3.4/bin:/usr/local/bin:/usr/bin:/bin:/usr/  
    sbin:/sbin:/opt/X11/bin:/Library/TeX/texbin  
andarilho: alessandrovivas$ echo $HOME  
/Users/alessandrovivas  
andarilho: alessandrovivas$ echo ~  
/Users/alessandrovivas
```

Quando utilizado com a opção **-e** pode ser utilizado com comandos especiais descritos na Tabela 1.2.

Atalho	Descrição da funcionalidade
\n	Salta Linhas (Listagem 1.17).
\t	Tabulação Horizontal (Listagem 1.18).
\v	Tabulação Vertical (Listagem 1.19).
\\\	Barra Invertida (Listagem 1.20).
\r	Retorno de Linha (Listagem 1.21).

Tabela 1.2: Atalhos para o Terminal

A Listagem 1.17 apresenta a sintaxe para saltar linhas com **echo**.

#### Listagem 1.17: Saltando Linhas com echo

```
andarilho: alessandrovivas$ echo -e  
Linha1  
Linha2
```

A Listagem 1.18 apresenta a sintaxe para realizar tabulação horizontal com **echo**.

#### Listagem 1.18: Tabulação Horizontal

```
andarilho: alessandrovivas$ echo -e  
Linha1 Linha2
```

A Listagem 1.19 apresenta a sintaxe para realizar tabulação vertical com **echo**.

#### Listagem 1.19: Tabulação Vertical

```
andarilho: alessandrovivas$ echo -e  
Linha1  
    Linha2
```

A Listagem 1.20 apresenta a sintaxe para imprimir barra invertida com **echo**.

#### Listagem 1.20: Barra Invertida

```
andarilho: alessandrovivas$ echo -e  
Linha1\Linha2
```

A Listagem 1.21 apresenta a sintaxe para retorno de linha com **echo**.

#### Listagem 1.21: Retorno de Linha

```
andarilho: alessandrovivas$ echo -e  
Linha1  
Linha2
```

---

## 1.10 Alterando o Nome da Máquina Temporariamente

Deseja alterar o nome de sua máquina? Para mudar temporariamente utilize o procedimento descrito na Listagem 1.22. O comando **sudo** indica que você irá realizar uma operação que somente o administrador pode realizar. Neste caso você irá necessitar de entrar com sua senha. O comando **hostname** serve tanto para visualizar o nome de sua máquina como também para alterar o nome da mesma.

#### Listagem 1.22: Alterando o Nome da Máquina

```
musashi:~ alessandrovivas$ sudo hostname vivas
```

---

A Listagem 1.23 apresenta a sintaxe para verificação do nome da máquina. Como pode ser visto é o mesmo comando utilizado para alterar o nome da máquina.

#### Listagem 1.23: Visualizando o Nome da Máquina

```
musashi:~ alessandrovivas$ hostname  
vivas
```

---

O terminal só atualizará o nome quando você abrir outro terminal. O resultado pode ser visualizado na Listagem 1.24.

#### Listagem 1.24: Resultado da Alteração do Nome da Máquina

```
Last login: Mon Nov 23 19:33:42 on ttys001  
vivas:~ alessandrovivas$
```

---

## 1.11 Alterando o Nome da Máquina Permanentemente

Para alterar permanentemente utilize o comando da Listagem 1.25. O nome só será atualizado nas seguintes condições: reinicialização da máquina ou abrir novamente o terminal. Neste exemplo o nome da máquina foi alterado permanentemente para o nome brasil utilizando o comando **hostname -s brasil**.

#### Listagem 1.25: Alteração Permanentemente

```
vivas:~ alessandrovivas$ sudo hostname -s brasil  
Password:
```

---

A Listagem 1.26 apresenta o comando para verificação se o nome foi alterado.

#### Listagem 1.26: Verificando a Alteração do Nome

```
vivas:~ alessandrovivas$ hostname  
brasil
```

---

## 1.12 Histórico do Terminal

Para facilitar as coisas o *Mac OS X* mantém o histórico dos comandos digitados, tanto válidos quanto inválidos. Isto evita que você fique perdendo tempo em digitar tudo novamente. Assim para navegar entre os últimos comandos passados ao sistema, utilize as setas direcionais ( $\uparrow$  ou  $\downarrow$ ). Ao apertar diversas vezes, tais comandos irão aparecer na ordem cronológica inversa, i.e., do mais recente para o mais antigo.

Quando se tem um histórico com poucos comandos a navegação por setas direcionais pode ser feita sem problemas. No entanto, quando a lista passa a contar com 50 ou mais comandos, a busca de um dado comando passa a ser enfadonha. Nesse caso, utilize o mecanismo de procura <**CONTROL+R**>, cuja interface é apresentada na Listagem 1.27.

### Listagem 1.27: Histórico

```
(reverse-i-search):  
// comece a digitar aquele comando find  
(reverse-i-search) fi: find / | grep a  
// basta digitar enter para executar o comando
```

Ao digitar o primeiro caractere, surgirá o comando mais recente que possui aquele caractere. Para refinar a seleção, deve-se continuar digitando outros caracteres e o comando mais próximo da seqüência digitada irá aparecer ao lado. Para executar a escolha reconhecida na busca basta apertar a tecla <**ENTER**>. Para editar o comando a tecla <**backspace**> deve ser usada.

Algumas vezes pode ser necessário editar algum comando do histórico antes de executá-lo. Para tanto você deverá utilizar as setas direcionais para direita ou esquerda ( $\leftarrow$  ou  $\rightarrow$ ) quando ver o comando desejado para editá-lo antes de executar.

### 1.12.1 Comando *history*

O comando **history** pode ser executado para listar o histórico de comandos utilizados no Terminal. A Listagem 1.28 apresenta o resultado da execução do comando.

### Listagem 1.28: Comando *history*

```
musashi:~ alessandrovivas$ history  
506 sudo -s hostname musashi  
507* /bin/ls  
508 ls /bin  
509 echo $SHELL  
510 echo $SHELL  
511 clear  
512 ls  
513 echo teste  
514 echo
```

Os comandos armazenados no histórico são apresentados em ordem cronológica e numerados. Para executar um dos comandos anteriores, basta utilizar exclamação e o número do comando. Por exemplo, para executar novamente o quarto comando, basta fazer

### Listagem 1.29: Executar um determinado comando do *history*

```
[avivas@musashi ~]$ !512  
ls
```

Para procurar um determinado comando no **history**, basta usá-lo em combinação com o comando **grep**. O exemplo abaixo na Listagem 1.30 ilustra o caso em que desejamos localizar um comando utilizado que contenha a palavra-chave **clear**.

### Listagem 1.30: Exemplo de busca no history

```
musashi:~ alessandrovivas$ history | grep clear
474  clear
476  clear
478  clear
487  clear
491  clear
492  clear
500  clear
501  clear
505  clear
511  clear
525  clear
527  clear
529  history | grep clear
```

---

Você pode limpar todo o histórico utilizando o comando exemplificado na Listagem 1.31.

### Listagem 1.31: Limpar o histórico

```
musashi:~ alessandrovivas$ history -c
```

---

O tamanho máximo do histórico é definido pela variável de ambiente **HISTSIZE**. Você poderá verificar o valor desta variável e modificá-lo, se julgar necessário, conforme exemplificado a seguir.

Para listar o valor atual da variável de ambiente que define o tamanho máximo utilize o código da Listagem 1.32.

### Listagem 1.32: Tamanho do histórico

```
musashi:~ alessandrovivas$ echo $HISTSIZE
500
```

---

# Desligando o Mac OS X

*Mas... para que isso serve?*

Robert Lloyd, IBM, falando sobre o microprocessador.

## Sumário

<b>2.1</b>	<b>Saindo do sistema</b>	<b>12</b>
<b>2.2</b>	<b>Desligando e Reiniciando o Sistema</b>	<b>12</b>
<b>2.3</b>	<b>Reiniciando a máquina</b>	<b>14</b>

## 2.1 Saindo do sistema

Para sair do terminal pode-se utilizar os comandos **exit** e **logout**.

### 2.1.1 Saindo do Sistema com Logout

Ao terminar seu trabalho você deve sair do sistema, para isto utilize o comando **logout**. A sintaxe é bastante simples e funciona quando você entrou no sistema via terminal, o procedimento é apresentado na Listagem 2.1.

#### Listagem 2.1: Comando logout

```
Last login: Tue Nov 24 06:52:25 on ttys001
musashi:LivroMac-1aEdicao alessandrovivass$ logout
3Saving session...
...copying shared history...
...saving history...truncating history files...
6...completed.
```

[Processo concluído]

---

### 2.1.2 Saindo do Sistema com Exit

Você pode também sair do terminal usando o comando **exit**. Apesar de praticamente iguais, o **exit** pode ser utilizado em qualquer script enquanto o **logout** não. O procedimento de uso do comando **exit** é apresentado Listagem 2.2.

#### Listagem 2.2: Comando exit

```
Last login: Tue Nov 24 06:43:54 on ttys000
musashi:LivroMac-1aEdicao alessandrovivass$ exit
3logout
Saving session...
...copying shared history...
6...saving history...truncating history files...
...completed.
Deleting expired sessions...39 completed.
9
[Processo concluído]
```

---

## 2.2 Desligando e Reiniciando o Sistema

Outra forma de sair do sistema é desligando a máquina. Nunca desligue a máquina sem os comandos apropriados, pois isto pode corromper o sistema de arquivos. Ao desligar a máquina com os comandos corretos, o *MAC OS X* finalizará os programas, gravará os dados no disco rígido e começará a mostrar procedimentos de finalização.

### 2.2.1 Desligando Imediatamente

Para desligar o computador utilizamos o comando **shutdown**. Este comando possui diversas variações e cabe ao usuário decidir qual a melhor opção.

Se quiser desligar imediatamente utilize o código da Listagem 2.3. Para executar este comando será necessário a senha do administrador e a opção **h** significa que é para parar o computador (**halt**).

### Listagem 2.3: Desligando Imediatamente com shutdown

```
musashi:~ alessandrovivas$ sudo shutdown -h now
Password:
```

## 2.2.2 Desligando após um determinado tempo

É possível programar para desligar o computador após qualquer intervalo de tempo. Suponha que você deseje desligar a máquina após 3 minutos e para isto é necessário passar o tempo como argumento como na Listagem 2.4.

### Listagem 2.4: Desligando Após Determinado Intervalo de Tempo

```
musashi:~ alessandrovivas$ sudo shutdown -h +3
Password:
3Shutdown at Tue Nov 24 07:01:09 2015.
shutdown: [pid 10746]
musashi:~ alessandrovivas$
6*** System shutdown message from alessandrovivas@musashi.local ***
System going down in 3 minutes
```

Se outra pessoa estiver logada no sistema irá receber mensagens parecidas com as da Listagem 2.5.

### Listagem 2.5: Mensagens recebidas

```
musashi:LivroMac-1aEdicao alessandrovivas$ 
*** System shutdown message from alessandrovivas@musashi.local ***
System going down in 3 minutes
```

## 2.2.3 Desligando em uma hora específica

Para desligar em uma hora determinada basta passar a hora desejada como argumento. A Listagem 2.6 apresenta o comando para desligar a máquina às 10:10 da manhã.

### Listagem 2.6: Desligando Imediatamente

```
$ sudo shutdown -h 10:10
```

## 2.2.4 Cancelando um shutdown

Quer interromper o comando de **shutdown**? Vamos supor que tenha digitado o seguinte comando da Listagem 2.7.

### Listagem 2.7: Desligando em 5 minutos

```
$ sudo shutdown -h +5
```

Após o envio do comando é necessário matar o processo de **shutdown**. Para isto é necessário descobrir o PID (Valor Numérico que Identifica o Processo no Sistema Operacional) como feito na Listagem 2.8.

### Listagem 2.8: Cancelando Shutdown

```
musashi:~ alessandrovivas$ ps -aef | grep shutdown
  0 10746      1  0  6:58    ??        0:00.00 shutdown -h +3
  3  501 10792 10686  0  6:59    ttys001   0:00.00 grep shutdown
```

O PID do processo é o número 10746 e agora será necessário utilizar o comando **kill** para matar o processo como na Listagem 2.9.

#### Listagem 2.9: Cancelando Shutdown

```
musashi:~ alessandrovivas$ sudo kill -9 10746
```

---

## 2.3 Reiniciando a máquina

Para reiniciar uma máquina podemos utilizar o comando **reboot** , Listagem 2.10. Outro comando que tem funcionalidade similar é o **halt** .

#### Listagem 2.10: Reiniciando com reboot

```
musashi:~ alessandrovivas$ sudo reboot  
Password:
```

---

O comando apresentado na Listagem 2.11 tem o mesmo resultado do comando **reboot**.

#### Listagem 2.11: Reiniciando com shutdown

```
musashi:~ alessandrovivas$ sudo reboot -r now  
Password:
```

---

Reiniciando após um determinado tempo, Listagem 2.12.

#### Listagem 2.12: Reiniciando após determinado tempo

```
musashi:~ alessandrovivas$ sudo reboot -r 3  
Password:
```

---

Reiniciando em uma hora específica, Listagem 2.13.

#### Listagem 2.13: Reiniciando em uma hora específica

```
musashi:~ alessandrovivas$ sudo reboot -r 10:10  
Password:
```

---

# Operações em Diretórios e Arquivos

*Nós nunca iremos desenvolver um sistema operacional de 32 bits.*

Bill Gates

## Sumário

3.1	Listando Arquivos e Diretórios . . . . .	16
3.2	Navegando em Diretórios . . . . .	19
3.3	Comando pwd . . . . .	20
3.4	Copiando Arquivos . . . . .	21
3.5	Copiando Múltiplos arquivos . . . . .	21
3.6	Copiando Diretórios e Sub-diretórios . . . . .	22
3.7	Movendo Arquivos . . . . .	22
3.8	Movendo e Renomeando Arquivos . . . . .	22
3.9	Criando um Arquivo Vazio com touch . . . . .	23
3.10	Apagando Arquivos . . . . .	23
3.11	Apagando um Diretório . . . . .	23
3.12	Apagando Diretório com rmdir . . . . .	23
3.13	Criando Diretório . . . . .	24
3.14	Trabalhando com Múltiplos Diretórios . . . . .	24

## 3.1 Listando Arquivos e Diretórios

O comando **ls** é muito flexível e permite ao usuário listar arquivos e pastas de diversas maneiras. Para listas simples utilize o comando da Seção 3.1.1, para listar com apenas uma coluna utilize a Seção 3.1.2, para o formato longo utilize a Seção 3.1.3, para obter informações dos diretórios sem imprimir o seu conteúdo vá para Seção 3.1.5, na Seção 3.1.6 apresenta o comando para listar arquivos ocultos, para classificar diretórios e arquivos utilize o comando da Seção 3.1.7, para organizar pelo tamanho em blocos utilize a Seção 3.1.8 e para listar recursivamente utilize a Seção 3.1.9.

### 3.1.1 Lista Simples

O comando **ls** é utilizado para listar arquivos e diretórios no *MAC OS X*. Ele possui várias opções de uso. O comando **ls**, Listagem 3.1, sem opções exibe o conteúdo na forma de uma lista, na Seção 3.1.4 são apresentados os parâmetros que identificam arquivos e diretórios

#### Listagem 3.1: Listando o Conteúdo de um Diretório

```
musashi:~ alessandrovivas$ ls
Applications
Bit-BookPRO Projects
Desktop
Dicionario
Documents
Downloads
Dropbox
Ebooks
Google Drive
// ocultamos o restante da lista
```

### 3.1.2 Listando uma única entrada por linha

O comando **ls -1** lista o conteúdo do diretório adicionando uma entrada (diretório e/ou arquivos) por linha. A Listagem 3.2 apresenta o resultado do comando.

#### Listagem 3.2: Uma Entrada por Linha

```
musashi:~ alessandrovivas$ ls -1
Applications
Bit-BookPRO Projects
Desktop
Dicionario
Documents
Downloads
Dropbox
Ebooks
Google Drive
...
```

### 3.1.3 Listando o Conteúdo no Formato Longo

Imprimir uma ocorrência por linha em formato longo, contendo as informações de cada arquivo (permissões, dono, data de criação ou modificação, tamanho, etc.), Listagem 3.3.

#### Listagem 3.3: Comando ls no formato longo

```
musashi:~ alessandrovivas$ ls -l
total 160
drwxr-xr-x    5 alessandrovivas  staff   170  2 Mai  2015 Applications
drwxr-xr-x    4 alessandrovivas  staff   136  2 Mai  2015 Bit-BookPRO Projects
drwx-----+  60 alessandrovivas  staff  2040 24 Nov 13:59 Desktop
drwxr-xr-x    3 alessandrovivas  staff   102  4 Set 14:57 Dicionario
drwx-----+  29 alessandrovivas  staff   986 23 Nov 19:27 Documents
drwx-----+  83 alessandrovivas  staff  2822 19 Nov 09:37 Downloads
drwx-----@  53 alessandrovivas  staff  1802 24 Nov 15:42 Dropbox
-rw-r--r--   1 alessandrovivas  staff    89 12 Mar  2015 oimundo.c
```

---

### 3.1.4 Informações sobre os arquivos e diretórios

Quando se utiliza o comando **ls** no formato longo aparecem várias informações associadas, **-** para indicar um arquivo, **d** para indicar um diretório e **l** para indicar um link. A Listagem 3.4 apresenta o resultado do comando **ls** no formato longo.

#### Listagem 3.4: Símbolo -

```
musashi:alessandrovivas$ ls -l /etc/hosts
-rw-r--r--  1 root  wheel  946  2 Mai  2015 /etc/hosts
```

---

Se for um diretório iremos encontrar o símbolo **d**, Listagem 3.5.

#### Listagem 3.5: Símbolo d

```
musashi:~ alessandrovivas$ ls -l /
total 60
3drwxrwxr-x+ 187 root  admin  6358 24 Nov 14:16 Applications
drwxrwxr-x    5 root  admin   170 12 Nov  2014 Incompatible Software
drwxr-xr-x    4 root  wheel   136 24 Out  2013 Install.4017wvEQc
6drwxr-xr-x+  67 root  wheel  2278  5 Nov 09:14 Library
drwxr-xr-x@  2 root  wheel    68  4 Out 16:45 Network
drwxr-xr-x@  4 root  wheel   136  5 Nov 09:18 System
9drwxr-xr-x   7 root  admin   238  4 Out 16:45 Users
```

---

Pode ser também um link quando a primeira letra é um **l**.

#### Listagem 3.6: Símbolo l

```
musashi:~ alessandrovivas$ ls -l /etc
lrwxr-xr-x@ 1 root  wheel   11  4 Out 16:42 /etc -> private/etc
```

---

### 3.1.5 Obtendo informações sobre diretórios

Se você deseja obter informações sobre um determinado diretório mas não quer listar o conteúdo, utilize a opção **ls -ld**. A Listagem 3.7 apresenta o resultado do comando aplicado ao diretório **/usr**.

#### Listagem 3.7: Obtendo informações sobre diretórios

```
musashi:~ alessandrovivas$ ls -ld /usr
drwxr-xr-x@ 11 root  wheel  374  4 Out 17:01 /usr
```

---

### 3.1.6 Listando Arquivos Ocultos

Exibir todos os arquivos, inclusive os arquivos escondidos (iniciados pelo ponto final .arq\_oculto), devemos utilizar o comando **ls -a** ou **ls --all**, Listagem 3.8.

**Listagem 3.8: Listando Todos os Arquivos Inclusive os Ocultos**

```
musashi:~ alessandrovivas$ ls -a
.
..
.CFUserTextEncoding
.Certisign
.DS_Store
.Rhistory
.Trash
.Xauthority
.adobe
.android
.bash_history
.bash_profile
.bash_profile.pysave
.bash_sessions
```

---

### 3.1.7 Classificando Arquivos e Diretórios

Classificar os arquivos (/ para diretórios, \* para executáveis, @ para links simbólicos, | para FIFOs e = para sockets), Listagem 3.9.

**Listagem 3.9: Classificando Arquivos e Diretórios**

```
musashi:~ alessandrovivas$ ls -lF
total 184
-rwxr-xr-x    1 alessandrovivas  staff   8432 24 Nov 17:00 teste*
drwx-----+   83 alessandrovivas  staff   2822 19 Nov 09:37 Downloads/
drwx-----@  53 alessandrovivas  staff   1802 24 Nov 15:42 Dropbox/
```

---

### 3.1.8 Imprimindo Informações sobre o Tamanho dos arquivos

Para imprimir os arquivos com o tamanho em blocos de 512 bytes. Nos sistemas Unix antigos os blocos eram de 512 bytes e por isto este comando ainda calcula desta forma. Para usar o comando use a Listagem 3.10.

**Listagem 3.10: Tamanho em Blocos**

```
musashi:~ alessandrovivas$ ls -s
total 184
64 file.pdf
24 oimundo
...
```

---

Para imprimir o tamanho em Kilobytes, Megabytes ou Terabytes utilize o código da Listagem 3.11.

**Listagem 3.11: Tamanho dos Arquivos**

```
musashi:~ alessandrovivas$ ls -lh
drwxr-xr-x    10 alessandrovivas  staff   340B  4 Jul 2013 src
-rwxr-xr-x     1 alessandrovivas  staff   8,2K 24 Nov 17:00 teste
-rw-r--r--     1 alessandrovivas  staff   78B   6 Nov 19:21 teste.c
-rw-r--r--     1 alessandrovivas  staff   768B 24 Nov 17:00 teste.o
```

```
-rw-r--r--      1 alessandrovivas  staff   2,6K 27 Out 19:37 teste.tex
-rw-r--r--      1 alessandrovivas  staff   438B 18 Dez 2013 teste.txt
-rw-r--r--      1 alessandrovivas  staff   1,7K 23 Dez 2013 teste1.txt
drwxrwxr-x     5 alessandrovivas  staff   170B 18 Dez 2013 ubuntu
...

```

Pode-se usar uma composição de opções **ls -alF**.

#### Listagem 3.12: Combinando Opções do Comando ls

```
musashi:~ alessandrovivas$ ls -alF
total 2280
drwxr-xr-x+  86 alessandrovivas  staff   2924 24 Nov 17:00 .
drwxr-xr-x    7 root         admin   238   4 Out 16:45 ..
-r-----  1 alessandrovivas  staff    8   4 Out 17:18 .CFUserTextEncoding
drwxr-xr-x@  2 alessandrovivas  staff   68  25 Fev 2013 .Certisign/
...

```

Algumas vezes você deseja visualizar o diretório e não o conteúdo do mesmo. Para isto, basta usar a opção **-d**:

#### Listagem 3.13: Visualizando Informações sobre o Diretório

```
musashi:~ alessandrovivas$ ls -ald
drwxr-xr-x+ 86 alessandrovivas  staff   2924 24 Nov 17:00 .
```

### 3.1.9 Listando Recursivamente

Para listar recursivamente o conteúdo de um diretório e seus sub-diretórios utilize o comando da Listagem 3.14. Para interromper o comando digite <**CONTROL+C**>.

#### Listagem 3.14: Listando Recursivamente

```
musashi:~ alessandrovivas$ ls -R
Applications
3Bit-BookPRO Projects
Desktop
Dicionario
Documents
Downloads
Dropbox
Ebooks
Google Drive
Library
12
...

```

## 3.2 Navegando em Diretórios

Esta Seção apresenta os comandos para navegação em diretórios. Na Seção 3.2.1 apresenta os comandos para entrar no diretório de trabalho

### 3.2.1 Diretório de Trabalho

Use o comando **cd** para a navegação nos diretórios do sistema de arquivo do **Mac OS X**. Para ir ao seu diretório **Users** (seu diretório padrão de trabalho), basta digitar o comando da Listagem 3.15. Se o usuário for pedro então

para ir para o diretório é necessário digitar o caminho completo. O comando para navegar nos diretórios é o **cd**

#### Listagem 3.15: Comando cd

```
musashi:~ alessandrovivas$ cd /Users/alessandrovivas/
```

Para ir rapidamente ao seu diretório **Users**, apenas digite:

#### Listagem 3.16: Atalho para o Diretório Raiz do Usuário

```
musashi:~ alessandrovivas$ cd ~
```

O símbolo **~** é expandido pelo interpretador como */Users/seu\_usuario* ao executar o comando. Para verificar isso, experimente:

#### Listagem 3.17: Significado de **~**

```
musashi:~ alessandrovivas$ echo ~  
/Users/alessandrovivas
```

Alternativamente pode ser usada uma forma reduzida que produzirá o mesmo efeito, ou seja, o comando **cd** sozinho, Listagem 3.18.

#### Listagem 3.18: Comando cd Sem Opções

```
musashi:/ alessandrovivas$ cd
```

Os diretórios **.** e **..** referem-se ao diretório corrente e diretório-pai, respectivamente. Esses diretórios podem ser usados em sintaxes de comandos, assim como o **~** também. Exemplos:

- **ls -la .**
- **ls -F ..**
- **cd ..**
- **cd ./dir1/dir2**

Para retornar para o diretório-pai basta usar o código da Listagem 3.19.

#### Listagem 3.19: Retornando ao Diretório do Usuário

```
$ cd ..
```

## 3.3 Comando **pwd**

Ao abrir o terminal automaticamente seu usuário é direcionado a algum diretório. Para saber qual é a sua localização atual, você poderá utilizar o comando **pwd** como na Listagem 3.20.

#### Listagem 3.20: Comando **pwd**

```
musashi:~ alessandrovivas$ pwd  
/Users/alessandrovivas
```

Todo diretório possui dois arquivos especiais cujos nomes consiste em um ou dois pontos: **'.'** ou **'..'**. Estes designam o diretório e o diretório pai, respectivamente.

Ao designar um arquivo, subentende-se que está sendo referenciado o arquivo no diretório corrente. Outra maneira é especificar o arquivo com o caminho completo, como ilustrado na Listagem 3.21.

#### Listagem 3.21: Outro Exemplo do Comando pwd

```
$ pwd  
/Users/john  
$ ls test.txt  
$ ls /home/john/test.txt  
$ ls ~/test.txt
```

---

## 3.4 Copiando Arquivos

Ao usar o comando **cp** (**copy**) , pode-se efetuar cópias de arquivo ou grupo de arquivos, bem como diretórios inteiros. O comando precisa de dois argumentos: o arquivo original e o destino. A sintaxe geral é apresentada na Listagem 3.22.

#### Listagem 3.22: Sintaxe do Comando cp

```
$ cp <opcoes> </localfonte/arquivo> </localdestino/>
```

---

### 3.4.1 Copiando Arquivo para Diretório

Para copiar um arquivo para um diretório utilize o comando da Listagem 3.23. neste caso o arquivo está no diretório corrente e será copiado para o diretório /home/pedro/documentos/livros.

#### Listagem 3.23: Copiando Arquivo para Diretório

```
musashi:~ alessandrovivas$ cp teste.c ~/documentos/Livros/
```

---

A barra final como indicação de diretório de destino é essencial. Caso não seja colocado, o sistema interpretará o último elemento do caminho de destino como sendo um nome de arquivo. Assim, ao copiar o arquivo teste.c (exemplo acima) ele teria seu nome alterado para seu\_usuário.

Suponha que esteja no diretório /Users/pedro/programas e deseja copiar o arquivo teste.c do diretório /Users-/pedro/aulas para /Users/pedro/testes. Para isto você vai precisar de passar o caminho completo do diretório ou o caminho relativo, como apresentado na Listagem 3.24.

#### Listagem 3.24: Copiando Arquivo para um Diretório

```
$ cp /Users/pedro/aulas/teste.c /Users/pedro/testes  
$ cp ../aulas/teste.c ../testes
```

---

## 3.5 Copiando Múltiplos arquivos

Use o comando **cp** para copiar múltiplos arquivos para um diretório, podendo estar tais arquivos em locais diferentes como apresentado na Listagem 3.25.

Neste exemplo copiamos os arquivos *code.c* no diretório */Users/pedro/dir1/* e o arquivo *main.c* no diretório */Users/pedro/dir2/* e finalmente o arquivo *teste.c* do diretório */Users/pedro/dir3/* para o diretório */Users/pedro/*. Pelo exemplo, podemos ver a flexibilidade do sistema, tendo o usuário total liberdade de definir quantos e quais arquivos devem ser copiados para o diretório de destino.

#### Listagem 3.25: Copiando Múltiplos Arquivos

```
$ cp /Users/pedro/dir1/code.c Users/pedro/dir2/main.c /Users/pedro/dir3/teste.c /Users/pedro/
```

---

Use o comando **cp** para copiar um arquivo para outro, Listagem 3.26. Dessa forma, o sistema não interpelará, i.e., caso haja um outro arquivo com o nome do arquivo a ser criado ele será sobreescrito sem nenhum impedimento. O sistema admitirá que você, usuário, sabe o que está fazendo. Para se efetuar a mesma ação com a necessidade de confirmação, você deve usar a opção **-i** para alertá-lo, caso seja necessário sobreescriver algum arquivo.

#### Listagem 3.26: Copiando um Arquivo em Outro

```
$ cp code.c main.c
```

---

A Listagem 3.27 apresenta o comando **cp** com confirmação (opção **-i**). Neste exemplo, cria o arquivo *main.c* caso não exista e lhe pede confirmação da ação caso seja necessário sobreescriver um arquivo já existente.

#### Listagem 3.27: Copiando com a Opção **-i**

```
$ cp -i code.c main.c  
cp: overwrite main.c?
```

---

## 3.6 Copiando Diretórios e Sub-diretórios

Use a opção **-r** para copiar hierarquias inteiras de arquivos e sub-diretórios(inclusive arquivos ocultos) como apresentado na Listagem 3.28. Neste caso iremos copiar o diretório */Users/maria/teste* para o diretório */Users/maria/temp*.

#### Listagem 3.28: Copiando Diretórios

```
$ cp -r /Users/maria/teste /Users/maria/temp
```

---

## 3.7 Movendo Arquivos

O comando **mv (move)** é utilizado para mover e renomear um ou mais arquivos. A sintaxe geral é apresentada na Listagem 3.29, através de um exemplo em que o arquivo *teste.txt* será movido para outro diretório.

#### Listagem 3.29: Movendo Arquivos com mv

```
$ mv teste.txt /Users/alessandrovivas/programas/teste2.txt
```

---

## 3.8 Movendo e Renomeando Arquivos

O comando **mv (move)** é utilizado para mover e renomear um ou mais arquivos. A sintaxe geral é apresentada na Listagem 3.30, através de um exemplo em que o arquivo *teste.txt* será renomeado para *teste2.txt*.

#### Listagem 3.30: Renomeando Arquivos com rename

```
$ mv teste.txt teste2.txt
```

---

## 3.9 Criando um Arquivo Vazio com touch

O comando **touch** é utilizado para criar um ou mais arquivos vazios. A sintaxe geral é apresentada na Listagem 3.31, através do exemplo que criará o arquivo vazio *teste.txt*.

### Listagem 3.31: Criando Arquivos com touch

```
$ touch teste.txt
```

---

## 3.10 Apagando Arquivos

Para apagar um arquivo utilizamos o comando **rm** como mostrado na Listagem 3.32. Vamos supor que precisamos apagar o arquivo *teste.c*.

### Listagem 3.32: Apagando um Arquivo

```
$ rm teste.c
```

---

### 3.10.1 Apagando Múltiplos Arquivos

Vamo supor que precisamos apagar todos os arquivos com a extensão **.c**. Utilize a Listagem 3.33.

### Listagem 3.33: Apagando Múltiplos Arquivos

```
$ rm *.c
```

---

## 3.11 Apagando um Diretório

Quer apagar um diretório inteiro? Vamos utilizar o comando **rm** como na Listagem 3.34. Vamos supor que precisamos apagar o sub-diretório *músicas*, para tanto, será necessário apagar os arquivos dentro deste diretório e o próprio diretório. Podemos passar o argumento **-r** ou **-R** para que sejam apagados recursivamente os arquivos e subdiretórios dentro do diretório que desejamos remover e também o próprio diretório a ser removido.

### Listagem 3.34: Apagando Diretório

```
$ rm -Rf músicas/
```

---

## 3.12 Apagando Diretório com rmdir

O comando **rmdir** pode ser utilizado para apagar diretórios como realizado na Listagem 3.35, onde é criado o diretório *teste* e em seguido é apagado.

### Listagem 3.35: Apagando Diretório com rmdir

```
mkdir teste  
rmdir teste
```

---

## 3.13 Criando Diretório

O comando **mkdir** (*make directory*) cria um ou mais sub-diretórios. A Listagem 3.36 apresenta o código de criação do diretório teste.

### Listagem 3.36: Criando Diretório com mkdir

```
$ mkdir teste
```

---

## 3.14 Trabalhando com Múltiplos Diretórios

### 3.14.1 Criando Múltiplos Diretórios

Quer criar vários diretórios de uma única vez. Utilize o comando **mkdir** como apresentado na Listagem 3.37. Neste exemplo são criados os diretórios *tempdir1*, *tempdir2* e *tempdir3*.

### Listagem 3.37: Criando Múltiplos Diretórios

```
$ mkdir tempdir1 tempdir2 tempdir3
```

---

### 3.14.2 Criar Hierarquia de Diretórios

Se você precisa criar uma hierarquia de sub-diretórios de uma única vez, basta usar a opção **-p** como na Listagem 3.38, desta forma todos os diretórios da hierarquia serão criados, caso não existam.

### Listagem 3.38: Criando Árvore de Diretórios

```
$ mkdir -p temp/temp1/tempdir
```

---

# Comandos para Manipulação Texto

*A televisão não deverá conseguir se  
segurar no mercado após seis meses.  
Ninguém vai querer ficar olhando para  
uma caixa de madeira todas as noites.*

Darryl Zanuck, fundador da 20th  
Century Fox, 1946

## Sumário

4.1 Comando apropos . . . . .	26
4.2 Comando cat . . . . .	26
4.3 Comparando Arquivos . . . . .	27
4.4 Comando csplit . . . . .	28
4.5 Cortando e Colando . . . . .	29
4.6 Inserindo e Retirando Tabulações . . . . .	30
4.7 Comando tr . . . . .	31
4.8 Comando fmt . . . . .	32
4.9 Comando grep . . . . .	33
4.10 Comando head . . . . .	33
4.11 Comando iconv . . . . .	33
4.12 Comando look . . . . .	34
4.13 Comando more . . . . .	34
4.14 Comando nl . . . . .	35
4.15 Comando pr . . . . .	35
4.16 Comando rev . . . . .	36
4.17 Comando stat . . . . .	36
4.18 Comando sort . . . . .	37
4.19 Comando tail . . . . .	37
4.20 Comando uniq . . . . .	38
4.21 Comando wc . . . . .	38

## 4.1 Comando apropos

O comando **apropos** busca na base de dados do comando **whatis** por uma determinada String. A Listagem 4.1 ilustra o procedimento de busca da String *disk*. O resultado será uma primeira página de comandos e para continuar pelas outras páginas utilize a tecla <ENTER>.

Listagem 4.1: Comando apropos

```
musashi:LivroMac-1aEdicao alessandrovivas$ apropos disk
FDERecoveryAgent(8)          - Full Disk Encryption Key Recovery Transmission Agent
3asr(8)                      - Apple Software Restore; copy volumes (e.g. from disk images)
autodiskmount(8)              - disk support tool
bitesize.d(1m)                - analyse disk I/O size by process. Uses DTrace
6bless(8)                     - set volume bootability and startup disk options
df(1)                         - display free disk space
diskarbitrationd(8)           - disk arbitration daemon
9diskhits(1m)                 - disk access by file offset. Uses DTrace
disklabel(8)                   - manipulate and query an Apple Label disk label
diskmanagementd(8)             - DiskManagement.framework server
12diskmanagementstartup(8)    - DiskManagement.framework helper tool
disktool(8)                    - disk support tool
diskutil(8)                    - modify, verify and repair local disks
15du(1)                        - display disk usage statistics
fdisk(8)                       - DOS partition maintenance program
hdid(8)                        - historical mechanism for attaching disk images
18hdik(8)                      - lightweight tool to attach and mount disk images in-kernel
hdutil(1)                      - manipulate disk images (attach, verify, create, etc)
hotspot.d(1m)                  - print disk event by location. Uses DTrace
21htcacheclean(8)              - Clean up the disk cache
iopattern(1m)                  - print disk I/O pattern. Uses DTrace
```

## 4.2 Comando cat

O comando **cat** é utilizado para apresentar no terminal um conteúdo de um arquivo e pode ser utilizado em conjunto com outros comandos. O arquivo *cidades.txt* possui o nome de todas as cidades do Brasil. Para visualizar o arquivo basta utilizar a sintaxe da Listagem 4.2.

Listagem 4.2: Comando cat

```
musashi:$ cat cidades.txt
RO Alta Floresta DOeste
3RO Alto Alegre dos Parecis
RO Alto Paraíso
RO Alvorada DOeste
6RO Ariquemes
RO Buritis
RO Cabixi
```

Se quiser visualizar um arquivo e inserir número da linha antes de cada linha impressa utilize a Listagem 4.3.

Listagem 4.3: Comando cat

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ cat -n cidades.txt
1 RO Alta Floresta DOeste
3 2 RO Alto Alegre dos Parecis
```

```
3 RO Alto Paraíso
4 RO Alvorada DOeste
6 5 RO Ariquemes
6 RO Buritis
7 RO Cabixi
9 8 RO Cacaualândia
9 RO Cacoal
10 RO Campo Novo de Rondônia
```

---

## 4.3 Comparando Arquivos

### 4.3.1 Comando cmp

Com o comando **cmp** é possível comparar dois arquivos e descobrir a diferença entre eles. Neste exemplo fizemos a cópia do arquivo *cidades.txt* para *cidades1.txt* no qual foi retirado apenas uma linha. A Listagem 4.4 apresenta as cinco linhas iniciais dos dois arquivos.

#### Listagem 4.4: Arquivos *cidades1.txt* e *cidades.txt*

```
musashi:LivroMac-1aEdicao alessandrovivas$ head -5 cidades.txt
RO Alta Floresta DOeste
3 RO Alto Alegre dos Parecis
RO Alto Paraíso
RO Alvorada DOeste
6 RO Ariquemes

9musashi:LivroMac-1aEdicao alessandrovivas$ head -5 cidades1.txt
RO Alta Floresta DOeste
RO Alto Alegre dos Parecis
12 RO Alto Paraíso
RO Alvorada DOeste
RO Buritis
```

---

O resultado do processamento do comando **cmp** é apresentado na Listagem 4.5.

#### Listagem 4.5: Arquivos *cidades1.txt* e *cidades.txt*

```
musashi:LivroMac-1aEdicao alessandrovivas$ cmp cidades.txt cidades1.txt
cidades.txt cidades1.txt differ: char 93, line 5
```

---

### 4.3.2 Comando comm

O comando **comm** compara dois arquivos ordenados e apresenta as diferenças entre os mesmos. A sintaxe do comando é apresentada na Listagem 4.6.

#### Listagem 4.6: Comando comm

```
$ cat comm01.txt
borracha
3caneta
réguia

6$ cat comm02.txt
caneta
```

```
clips
9grampeador

$ comm comm01.txt comm02.txt
12borracha
caneta
clips
15      grampeador

réguia
```

O resultado será apresentado em três colunas. Na primeira coluna estão as ocorrências que aparecem apenas no primeiro arquivo, na segunda coluna as ocorrências que aparecem apenas no segundo arquivo e na terceira coluna aparecem as ocorrências que estão nos dois arquivos.

## 4.4 Comando csplit

O comando **csplit** é utilizado para dividir um arquivo em outras unidades menores de acordo com um determinado padrão. O arquivo *cidades.txt* possui o nome de todas as cidades do Brasil. Sem argumentos ele pega um arquivo e divide de acordo com o número da linha. O arquivo original tem 5565 linhas e como o parâmetro passado é 1000 teremos dois arquivos: *xx00*(1000 linhas) e *xx01* (com o restante das linhas). A sintaxe da Listagem 4.7.

### Listagem 4.7: Comando csplit

```
musashi:LivroMac-1aEdicao alessandrovivas$ csplit cidades.txt 1000
17263
372409
```

Assim, ele dividiu nosso arquivo em dois outros *xx00* e *xx01*. A peça de Hamlet tem 5 atos e cada ato é separado pelo identificador *ACT*. Se você quiser separar cada ato em um arquivo diferente basta usar expressões regulares como na Listagem 4.8. Ele irá procurar 5 ocorrências da palavra *ACT* e criar os arquivos todos com o início *act*.

### Listagem 4.8: Comando csplit

```
musashi:LivroMac-1aEdicao alessandrovivas$ csplit -s -f act. hamlet.txt /ACT/ {3}
3musashi:LivroMac-1aEdicao alessandrovivas$ ls act*
act.00  act.01  act.02  act.03  act.04

6musashi:LivroMac-1aEdicao alessandrovivas$ head -2 act*
==> act.00 <==
ACT I
9

==> act.01 <==
12ACT II

15==> act.02 <==
ACT III

18
==> act.03 <==
ACT IV
21

==> act.04 <==
```

## 4.5 Cortando e Colando

### 4.5.1 Comando cut

Precisa cortar caracteres de uma frase? Vamos utilizar o comando **cut**. Neste exemplo, Listagem 4.9, vamos imprimir somente os caracteres 1 até 10 do nome "alessandro vivas andrade".

#### Listagem 4.9: Comando cut

```
musashi:~ alessandrovivas$ echo alessandro | cut -c 1-10
```

Outro problema recorrente é separação de valores em um arquivo. Na Listagem 4.10 temos um arquivo com estado com dois caracteres e após o nome da cidade. Neste arquivo estão todos os municípios brasileiros.

#### Listagem 4.10: Arquivo com Cidades

```
SP Sao Paulo  
SP Campinas  
MG Belo Horizonte  
MG Diamantina  
MG Lavras  
MG Bom Sucesso  
RJ Rio de Janeiro
```

Partindo da Listagem 4.10 vamos imprimir somente os dois primeiros caracteres de cada linha de um arquivo, Listagem 4.11.

#### Listagem 4.11: Separando dados de um Arquivo

```
musashi:alessandrovivas$ cat cidades.txt | cut -c 1-2  
RO  
RO  
RO  
RO  
RO  
RO  
RO
```

### 4.5.2 Comando paste

O comando **paste** é interessante para unir arquivos diferentes formatando as colunas. Imagine que eu tenha separado em dois arquivos, *nomes.txt* com os nomes dos alunos e *notas.txt* com as notas dos alunos. A Listagem 4.12 mostra os dois arquivos.

#### Listagem 4.12: Arquivos de Exemplo para o Comando paste

```
musashi:$ cat nomes.txt  
victor  
luciana  
pedro  
andre
```

```
musashi:alessandrovivas$ cat notas.txt
10 20
30 40
50 60
70 80
```

---

Vamos supor que agora eu necessite juntar os dois arquivos em colunas como na Listagem 4.13.

#### Listagem 4.13: Combinando Dois Arquivos com o Comando paste

```
musashi:alessandrovivas$ paste nomes.txt notas.txt
victor 10 20
luciana 30 40
pedro 50 60
andre 70 80
```

---

## 4.6 Inserindo e Retirando Tabulações

### 4.6.1 Comando expand

Se você estiver trabalhando com um arquivo com tabulações muitas vezes é necessário converter estas tabulações em espaços. Para realizar esta tarefa utilize o comando **expand**. A Listagem 4.14 apresenta os dados originais, armazenados no arquivo *cidades.txt*.

#### Listagem 4.14: Listagem Original

```
musashi:alessandrovivas$ cat cidades.txt
RO Alta Floresta DOeste
RO Alto Alegre dos Parecis
RO Alto Paraíso
RO Alvorada DOeste
RO Ariquemes
```

---

Convertendo a tabulação para 1 caractere na Listagem 4.15.

#### Listagem 4.15: Tabulações Convertidas para 1 Espaço

```
musashi:alessandrovivas$ cat cidades.txt | expand -t 1
RO Alta Floresta DOeste
RO Alto Alegre dos Parecis
RO Alto Paraíso
RO Alvorada DOeste
RO Ariquemes
```

---

Convertendo a tabulação para 5 caracteres na Listagem 4.16.

#### Listagem 4.16: Convertendo Tabulações

```
musashi:alessandrovivas$ cat cidades.txt | expand -t 5
RO Alta Floresta DOeste
RO Alto Alegre dos Parecis
RO Alto Paraíso
RO Alvorada DOeste
RO Ariquemes
```

---

Podemos fazer este tipo de substituição utilizando um comando mais genérico, o comando **tr**, como veremos em seguida.

## 4.6.2 Comando unexpand

Imagine que agora nosso arquivo não tenha tabulações, Listagem 4.17 separando os campos e precisamos converter os espaços em tabulação. Para realizar esta tarefa utilize o comando **unexpand**. A Listagem 4.17 apresenta os dados originais, armazenados no arquivo *cidades.txt*.

### Listagem 4.17: Listagem Original

```
musashi:alessandrovivas$ cat cidades1.txt
RO Alta Floresta DOeste
RO Alto Alegre dos Parecis
RO Alto Paraíso
RO Alvorada DOeste
RO Ariquemes
```

---

Convertendo os 5 espaços para tabulação como mostrado na Listagem 4.18.

### Listagem 4.18: Convertendo Espaços em Tabulações

```
musashi:alessandrovivas$ cat cidades1.txt | unexpand -t 5
RO Alta Floresta DOeste
RO Alto Alegre dos Parecis
RO Alto Paraíso
RO Alvorada DOeste
RO Ariquemes
```

## 4.7 Comando tr

O comando **tr** é utilizado para efetuar substituições (ou tradução) e apagar caracteres. A Listagem 4.19 apresenta um exemplo em que utilizaremos o comando **tr** para substituir tabulações por uma vírgula.

### Listagem 4.19: Convertendo tabulações em espaço simples

```
musashi:alessandrovivas$ cat cidades.txt | tr '\t' ','
RO,Alta Floresta DOeste
RO,Alto Alegre dos Parecis
RO,Alto Paraíso
RO,Alvorada DOeste
RO,Ariquemes
```

---

Se além disso, queremos substituir as múltiplas ocorrências de espaços por um único espaço, podemos proceder como ilustrado na Listagem 4.20.

### Listagem 4.20: Convertendo tabulações em espaço simples e removendo múltiplas ocorrências de espaços

```
musashi:alessandrovivas$ cat cidades.txt | tr '\t' ' ' | tr -s ' '
RO Alta Floresta DOeste
RO Alto Alegre dos Parecis
RO Alto Paraíso
RO Alvorada DOeste
RO Ariquemes
```

---

O comando **tr** pode ser utilizado para realizar diversos outros tipos de substituições ou para apagar caracteres indesejáveis. Veremos abaixo alguns exemplos.

### Listagem 4.21: Convertendo maiúsculas em minúsculas

```
musashi:alessandrovivas$ cat cidades.txt | tr 'A-Z' 'a-z'  
ro alta floresta doeste  
ro alto alegre dos parecis  
ro alto paraíso  
ro alvorada doeste  
ro ariquemes
```

---

Uma outra forma de realizar a substituição de maiúsculas por minúsculas é apresentada na Listagem 4.22.

#### Listagem 4.22: Outra forma de converter MAIÚSCULA em minúsculas

```
$ tr [:upper:] [:lower:]
```

---

O exemplo da Listagem 4.23 ilustra como transformar espaços em branco (incluindo aqui tabulações e quebras de linhas) em uma quebra de linha. Para tanto, utilizaremos [ :space:] para designar qualquer um dos caracteres: espaço, tabulação e quebra de linha.

#### Listagem 4.23: Transformar espaços em quebra de linha

```
$ tr -s [:space:] | tr [:space:] '\n'
```

---

Podemos utilizar o comando **tr** para substituir um conjunto de caracteres. Para tanto, será considerada a ordem em que eles parecem. No exemplo apresentado na Listagem 4.24 iremos substituir { por ( e } por ).

#### Listagem 4.24: Substituir chaves por parênteses

```
$ tr '{}()'()
```

---

Caso deseje remover os algarismos de 0 a 9, basta utilizar uma das duas formas ilustradas na Listagem 4.25.

#### Listagem 4.25: Duas maneiras para se remover dígitos

```
$ tr -d [:digit:]  
$ tr -d '0-9'
```

---

## 4.8 Comando fmt

O comando **fmt** é usado para formatar arquivos texto. Usado para organizar as palavras(grupos de caracteres) de um arquivo para uma forma consistente, i.e., com um número de caracteres por linha definido.

Seja o seguinte texto, sem delimitação de caracteres: "Antes da chegada dos colonizadores portugueses, no século XVI (os primeiros relatos dão conta de expedições que subiram o Rio Jequitinhonha e São Francisco), Diamantina, como toda a região do atual estado de Minas Gerais, era ocupada por povos indígenas do tronco linguístico ". Vamos supor que precisamos formatar o texto e limitar 10 caracteres por linha. Para fazer isto utilizamos o comando **fmt** da Listagem 4.26.

#### Listagem 4.26: Formatando Linhas com o Comando fmt

```
musashi: alessandrovivas$ fmt -w10 texto.txt  
Antes da  
chegada  
dos  
colonizadores  
portugueses,  
no  
...  
continua
```

---

## 4.9 Comando grep

O comando **grep** pode ser utilizado para procurar padrões em arquivos texto. Ele pode ser utilizado sozinho ou em conjunto com outros comandos. Vamos usar a Listagem 4.27 como exemplo. Suponha que você deseja encontrar em um arquivo um determinado padrão, como por exemplo, a palavra Sucesso no arquivo que contém o nome de todas as cidades do Brasil.

### Listagem 4.27: Listagem para uso do grep

```
musashi:alessandrovivas$ cat cidades.txt | grep Sucesso
PB  Bom Sucesso
MG  Bom Sucesso
SP  Bom Sucesso de Itararé
PR  Bom Sucesso
PR  Bom Sucesso do Sul
```

O comando `|` é chamado de pipe e é utilizado para concatenar um ou mais comandos.

## 4.10 Comando head

O comando **head** é utilizado para imprimir os `n` linhas iniciais de um arquivo. Imagine um arquivo com o nome de *cidades.txt* com todas as cidades do Brasil. A Listagem 4.28 apresenta a lista impressa do arquivo *cidades.txt*.

### Listagem 4.28: Imprime as Linhas Iniciais de um Arquivo

```
musashi:alessandrovivas$ head cidades.txt
RO Alta Floresta DOeste
RO Alto Alegre dos Parecis
RO Alto Paraíso
RO Alvorada DOeste
RO Ariquemes
RO Buritis
RO Cabixi
RO Cacaualândia
RO Cacoal
RO Campo Novo de Rondônia
```

Para imprimir as `N` linhas iniciais utilize o comando da Listagem 4.29. Neste exemplo iremos imprimir as duas linhas iniciais.

### Listagem 4.29: Imprime as Duas Linhas Iniciais de um Arquivo

```
musashi:alessandrovivas$ head -2 cidades.txt
RO Alta Floresta DOeste
RO Alto Alegre dos Parecis
```

## 4.11 Comando iconv

O comando **iconv** é utilizado para realizar conversões de caracteres, isto é, pegar uma string, texto ou arquivo codificado em um padrão de caracteres para outro padrão. Este comando pode ser útil, por exemplo, quando temos

um arquivo texto feito no Windows e desejamos utilizá-lo no **Mac OS X**. A Listagem 4.30 ilustra o exemplo do comando **iconv**.

#### Listagem 4.30: Convertendo Padrões de Caracteres

```
$ iconv -f iso-8859-1 -t utf-8 cidades.txt > cidadeutf8.txt
```

---

## 4.12 Comando look

O comando **look** é utilizado para visualizar linhas que possuem uma determinada string. Na Listagem 4.31 vamos procurar no arquivo *cidades.txt* as linhas que possuem a string dog.

#### Listagem 4.31: Comando look

```
musashi:alessandrovivas$ look dog
dog
dogal
dogate
dogbane
Dogberry
dogberry
Dogberrydom
Dogberryism
dogbite
dogblow
dogboat
dogbolt
dogbush
dogcart
```

---

Se nenhum arquivo for especificado, o comando **look** procura ocorrências no arquivo /usr/share/dict/words como na Listagem 4.32.

#### Listagem 4.32: Comando look

```
musashi:alessandrovivas$ look dog
dog
dogal
dogate
dogbane
Dogberry
dogberry
Dogberrydom
Dogberryism
dogbite
dogblow
dogboat
dogbolt
dogbush
dogcart
```

---

## 4.13 Comando more

O comando **more** é utilizado para processar e visualizar arquivos longos que não cabem na tela do terminal. Ele pode ser utilizado em conjunto com outros comandos. O arquivo *cidades.txt* é um arquivo muito grande e uma opção de visualização é usar o **more** como na Listagem 4.33. Ele coloca as informações suficientes para ocupar o tamanho da tela. Para navegar no arquivo basta apertar espaço ou o número da página após os dois pontos. Para sair digite q.

#### Listagem 4.33: Visualizando Arquivos Longos

```
musashi:alessandrovivas$ cat cidades.txt | more
RO Alta Floresta DOeste
RO Alto Alegre dos Parecis
RO Alto Paraíso
RO Alvorada DOeste
RO Ariquemes
RO Buritis
RO Cabixi
RO Cacaualândia
RO Cacoal
RO Campo Novo de Rondônia
RO Candeias do Jamari
RO Castanheiras
RO Cerejeiras
RO Chupinguaia
RO Colorado do Oeste
RO Corumbiara
RO Costa Marques
RO Cujubim
RO Espigão DOeste
RO Governador Jorge Teixeira
RO Guajará-Mirim
RO Itapuã do Oeste
RO Jaru
:
```

## 4.14 Comando nl

O comando **nl** conta o número de linhas de um arquivo. A Listagem 4.34 apresenta o uso do comando processando o arquivo *cidades.txt*.

#### Listagem 4.34: Contando o Número de Linhas

```
musashi:alessandrovivas$ nl cidades.txt
 1 RO Alta Floresta DOeste
 2 RO Alto Alegre dos Parecis
.....
5562 GO Vicentinópolis
5563 GO Vila Boa
5564 GO Vila Propício
5565 DF Brasília
```

## 4.15 Comando pr

O comando **pr** pega o conteúdo de um arquivo, divide o arquivo em colunas e páginas. Na Listagem 4.35 o conteúdo do arquivo *cidades.txt* foi dividido em 4 colunas.

#### Listagem 4.35: Dividindo em Colunas

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ cat cidades.txt | pr -4
```

```
1 Dez 07:06 2015 Page 1
```

RO Alta Flo AC	Capixaba AM	Manicor? PA	Barcaren
RO Alto Ale AC	Cruzeiro AM	Maraã PA	Belém
RO Alto Par AC	Epitacio AM	Maués PA	Belterra
RO Alvorada AC	Feijó AM	Nhamund? PA	Benevide
RO Ariqueme AC	Jordão AM	Nova Oli PA	Bom Jesu
RO Buritis AC	Mâncio AM	Novo Air PA	Bonito
RO Cabixi AC	Manoel U AM	Novo Ari PA	Braganç
RO Cacaúlã AC	Marechal AM	Parintin PA	Brasil N
RO Cacoal AC	Plácido AM	Pauini PA	Brejo Gr
RO Campo No AC	Porto Ac AM	Presiden PA	Breu Bra
RO Candeias AC	Porto Wa AM	Rio Pret PA	Breves
RO Castanhe AC	Rio Bran AM	Santa Is PA	Bujaru
RO Cerejeir AC	Rodrigue AM	Santo An PA	Cachoeir
RO Chupingu AC	Santa Ro AM	São Gab PA	Cachoeir
RO Colorado AC	Sena Mad AM	São Pau PA	Cametá
RO Corumbia AC	Senador AM	São Seb PA	Canaã d
RO Costa Ma AC	Tarauac? AM	Silves PA	Capanema
RO Cujubim AC	Xapuri AM	Tabating PA	Capitão
RO Espigão AM	Alvarãe AM	Tapauá PA	Castanha
...			

## 4.16 Comando rev

O comando **rev** inverte os caracteres de uma linha e envia o resultado para a saída padrão. Bastante útil para a manipulação de caracteres e programas de codificação. Ele recebe como argumento um arquivo ou uma mensagem. A Listagem 4.36 mostra a sintaxe para passar um arquivo como argumento. Neste comando utilizamos o arquivo *nomes.txt* da Listagem 4.12.

#### Listagem 4.36: Comando para Inverter os Caracteres - rev

```
musashi:alessandrovivas$ cat nomes.txt | rev
rotciv
zanaicul
ordep
erdna
```

A Listagem 4.37 apresenta a sintaxe para passar uma frase como argumento.

#### Listagem 4.37: Invertendo Caracteres com rev

```
musashi:alessandrovivas$ echo teste | rev
etset
```

## 4.17 Comando stat

O comando **stat** apresenta informações sobre um arquivo. A Listagem 4.38 apresenta o resultado do comando.

### Listagem 4.38: Comando stat

```
musashi:LivroMac-1aEdicao alessandrovivas$ stat cidades.txt
16777220 30040506 -rw-r--r-- 1 alessandrovivas staff 0 89672 Nov 25 15:02:22 2015 Nov 25
11:13:07 2015 Nov 25 11:13:07 2015 Nov 25 11:13:07 2015 4096 176 0 cidades.txt
```

---

## 4.18 Comando sort

Para ordenar arquivos utilizamos o comando **sort**. Seja um arquivo *nomes.txt* com os seguintes itens na Listagem 4.39.

### Listagem 4.39: Ordenando Arquivos com sort

```
musashi:LivroMac-1aEdicao alessandrovivas$ sort nomes.txt
victor
luciana
pedro
andre
```

---

Para ordenar o arquivo *teste.txt* vamos utilizar o comando **sort** na Listagem 4.40.

### Listagem 4.40: Ordenando Arquivos com sort

```
musashi:alessandrovivas$ cat nomes.txt | sort -n
andre
luciana
pedro
victor
```

---

## 4.19 Comando tail

O comando **tail** é utilizado para imprimir as n linhas finais de um arquivo. Imagine um arquivo com o nome de *cidades.txt* com todas as cidades do Brasil. A Listagem 4.41 apresenta a lista impressa do arquivo *cidades.txt*.

### Listagem 4.41: Imprime as Linhas Finais de um Arquivo

```
musashi:alessandrovivas$ tail cidades.txt
GO Uruaçu
GO Uruana
GO Urutai
GO Valparaíso de Goiás
GO Varjão
GO Vianópolis
GO Vicentinópolis
GO Vila Boa
GO Vila Propício
DF Brasília
```

---

Para imprimir as N linhas finais utilize o comando da Listagem 4.42. Neste exemplo iremos imprimir as duas linhas finais.

#### Listagem 4.42: Imprime as Duas Linhas Finais de um Arquivo

```
musashi:alessandrovivas$ tail -2 cidades.txt
GO Vila Boa
GO Vila Propício
DF Brasília
```

## 4.20 Comando uniq

O comando **uniq** é usado para encontrar linhas únicas num arquivos, i.e., ele remove linhas duplicadas consecutivas contidas em arquivos. É importante que o arquivo já esteja organizado para que ele possa remover todas as duplicações. Geralmente esse comando trabalha em conjunto com o sort. Ele possui as seguintes opções :

- **-c** : conta quantas vezes cada linha apareceu
- **-u**: imprime somente as linhas únicas
- **-d**: imprime somente linhas duplicadas

Vamos supor que você quer saber quantas palavras distintas existem em uma lista de palavras. Vamos utilizar a combinação dos seguintes comandos:

- **sort** : ordena as palavras
- **uniq**: retira frases com palavras iguais
- **wc**: conta as palavras

A Listagem 4.43 ilustra o comando

#### Listagem 4.43: Comando uniq

```
musashi:alessandrovivas$ sort /usr/share/dict/words | uniq | wc -l
235886
```

Caso deseje contar as palavras em um texto, por exemplo, em Hamlet de Shakespeare. Vamos utilizar os comandos vistos anteriormente para realizar esta tarefa. Primeiramente vamos substituir todas maiúsculas por minúsculas, em seguida vamos remover todos os caracteres que não estiverem entre a-z e também não forem espaço (espaço em branco, tabulação, quebra de linha). Feito isso, iremos substituir todo espaço por quebra de linha, em seguida ordenar as palavras, contabilizar apenas uma ocorrência de cada palavra e por fim contar quantas linhas foram geradas, ou seja, quantas são as palavras (tipos) utilizadas no texto. O código utilizado está ilustrado na Listagem 4.44.

#### Listagem 4.44: Contando quantas palavras distintas existem em um texto

```
musashi:alessandrovivas$ cat hamlet.txt | tr 'A-Z' 'a-z' | tr -dc 'a-z[:space:]' | tr [:space:] '\n' | sort | uniq | wc -l
4763
```

## 4.21 Comando wc

O comando **wc** conta o número de linhas, palavras e bytes (ou caracteres) de um arquivo. A Listagem 4.45 apresenta o uso do comando processando o arquivo *cidades.txt*.

#### Listagem 4.45: Contando o Número de Caracteres

```
musashi:alessandrovivas$ wc cidades.txt
      5565    15831    89672  cidades.txt
```

---

O comando imprime o número de linhas (5593), palavras (15850) e caracteres (89672). Para contar apenas o número de linhas utilize o comando **wc -l** como na Listagem 4.46.

#### Listagem 4.46: Contando o Número de Linhas

```
musashi:alessandrovivas$ wc -l cidades.txt
      5565  cidades.txt
```

---

Para contar o número de palavras utilize o comando **wc -w** como na Listagem 4.47.

#### Listagem 4.47: Contando o Número de Palavras

```
musashi:alessandrovivas$ wc -w cidades.txt
      15831  cidades.txt
```

---

Para contar o número de bytes utilize o comando **wc -c** como na Listagem 4.48, ou utilize, **wc -m** para contar o número de caracteres.

#### Listagem 4.48: Contando o Número de Bytes

```
musashi:alessandrovivas$ wc -m cidades.txt
      89672  cidades.txt
```



# Comandos de Sistema

*Eu acredito que talvez há no mercado espaço para uns cinco computadores.*

Thomas Watson, IBM, em 1943

## Sumário

5.1 Comando alias . . . . .	42
5.2 Comando basename . . . . .	42
5.3 Shell builtin . . . . .	42
5.4 Gerando Todos os Comandos com compgen . . . . .	42
5.5 Comando complete . . . . .	43
5.6 Comando cron . . . . .	43
5.7 Quem sou eu e onde estou? . . . . .	44
5.8 Comando id . . . . .	46
5.9 Alterando a Senha . . . . .	47
5.10 Usuários Logados . . . . .	47
5.11 Comandos de Calendário, Data e Hora . . . . .	47
5.12 Comando chflags . . . . .	49
5.13 Comando finger . . . . .	49
5.14 Comando last . . . . .	50
5.15 Comando vm_stat . . . . .	50
5.16 Comando su . . . . .	51
5.17 Comando uname . . . . .	51
5.18 Comando uptime . . . . .	52
5.19 Variável PATH . . . . .	52
5.20 Comando w . . . . .	53
5.21 Comando whereis . . . . .	53
5.22 Comando which . . . . .	53
5.23 Procurando Informações . . . . .	53
5.24 Rodando Comandos . . . . .	54

## 5.1 Comando alias

O comando **alias** permite criar apelidos para comandos. Vamos supor que você queira que a palavra lista seja equivalente ao comando **ls -l**. A Listagem 5.1 apresenta o resultado.

### Listagem 5.1: Criando Apelidos para Comandos

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ alias lista='ls -l'

% resultado
musashi-2:LivroMac-1aEdicao alessandrovivas$ lista
total 136880
-rwxr-xr-x@ 1 alessandrovivas  staff      1362100 16 Mar  2013 Caracara01.JPG
-rwxr-xr-x@ 1 alessandrovivas  staff      1320772 16 Mar  2013 Caracara02.JPG
-rwxr-xr-x@ 1 alessandrovivas  staff      1357615 16 Mar  2013 Caracara03.JPG
...
```

## 5.2 Comando basename

O comando **basename** permite retirar as barras de strings ou diretório base de uma string. A Listagem 5.2 apresenta o resultado.

### Listagem 5.2: Comando basename

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ basename /usr/bin/
bin
```

## 5.3 Shell builtin

Um comando é **builtin** quando ele pertence ao shell que você está executando. Quando este comando pertence ao shell ele geralmente roda na memória principal e assim tem seu desempenho melhorado. A Listagem 5.3 mostra a sintaxe para verificar se o comando é **builtin**. Para isto é necessário utilizar o comando **type**.

### Listagem 5.3: Comando builtin

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ type pwd
pwd is a shell builtin
```

Já o comando **whoami** não é builtin, Listagem 5.4.

### Listagem 5.4: Comando whoami

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ type whoami
whoami is /usr/bin/whoami
```

## 5.4 Gerando Todos os Comandos com **compgen**

Se você deseja saber todos os comandos incluídos em sua distribuição basta digitar o comando **compgen**. Para isto utilize a opção **-c** como na Listagem 5.5.

#### Listagem 5.5: Opções Múltiplas

```
musashi:LivroMac-1aEdicao alessandrovivass$ compgen -c
if
then
else
elif
fi
case
esac
for
select
while
until
do
done
in
function
time
{
}
!
[[ 
]]
...
alias
bg
bind
break
builtin
caller
cd
command
compgen
complete
continue
```

## 5.5 Comando complete

O comando **complete** é um comando bash para fazer a mesma função de quando usamos o TAB duas vezes.

## 5.6 Comando cron

O comando **cron** é utilizado para agendar a execução de serviços sejam periódicos ou não. Primeiro defina qual a tarefa que irá ser realizada.

Para agendar uma tarefa digite o comando **crontab** da Listagem 5.7. Insira os dados separados por tabulação, os campos são definidos como:

- 13: minutos
- 19: horas

- 2 : dia do mês
- \* : Qual mês do ano (neste caso todos os meses)
- \*: Dia da semana (0 - Domingo, 6 - Sábado)
- ls > crontab.txt: comando ou script a ser executado

#### Listagem 5.6: Agendamento com crontab

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ crontab -e
13      19      2      *      *      ls > crontab.txt
```

Quando digitar todos os dados salve com o comando :w e para sair digite :q. Para listar os agendamentos digite o comando da Listagem 5.7.

#### Listagem 5.7: Listando os Agendamentos

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ crontab -l
13 19 2 * * ls > crontab.txt
```

## 5.7 Quem sou eu e onde estou?

Se você tem problemas de múltiplas personalidades esta é uma boa opção. Brincadeiras a parte, estes comandos são muito importantes no uso diário.

- Quem sou eu? Para saber quem é você (seu username) utilize o comando **whoami** . Este comando é muito utilizado para saber com que usuário você está logado. Às vezes fica muito confuso quando somos o superusuário, ou quando utilizamos mais de um usuário no sistema.
- Onde estou? Para saber o local em que você se encontra na árvore de diretórios (diretório de trabalho ou diretório corrente) use o comando **pwd** .

A Listagem 5.8 apresenta o resultado dos comandos.

#### Listagem 5.8: Comando whoami e pwd

```
musashi:LivroMac-1aEdicao alessandrovivas$ whoami
alessandrovivas
musashi:LivroMac-1aEdicao alessandrovivas$ pwd
/Users/alessandrovivas/Documents/Livros/LivroMac-1aEdicao
```

### 5.7.1 Arquivo dscl

O **Mac OS X** não utiliza o arquivo **passwd** para armazenar os usuários do sistema. Para listar todos os usuários utilizamos o comando **dscl** . A Listagem 5.9 lista todos os usuários do sistema.

#### Listagem 5.9: Listando Usuários

```
musashi:LivroMac-1aEdicao alessandrovivas$ dscl . -list /Users
_amavisd
_appleevents
_appowner
_appserver
_ard
_assetcache
_astris
```

\_atsserver  
\_avbdeviced  
\_calendar  
\_ces  
\_clamav  
\_coreaudiod  
\_coremediaiod  
\_cvmsroot  
\_cvs  
\_cyrus  
\_devdocs  
\_devicemgr  
\_displaypolicyd  
\_distnote  
\_dovecot  
\_dovenull  
\_dpaudio  
\_eppc  
\_ftp  
\_gamecontrollerd  
\_geod  
\_iconservices  
\_installassistant  
\_installer  
\_jabber  
\_kadmin\_admin  
\_kadmin\_changepw  
\_krb\_anonymous  
\_krb\_changepw  
\_krb\_kadmin  
\_krb\_kerberos  
\_krb\_krbtgt  
\_krbfast  
\_krbtgt  
\_launchservicesd  
\_lda  
\_locationd  
\_lp  
\_mailman  
\_mbsetupuser  
\_mcxalr  
\_mdnsresponder  
\_mysql  
\_netbios  
\_netstatistics  
\_networkd  
\_nsurlsessiond  
\_nsurlstoraged  
\_ondemand  
\_postfix  
\_postgres  
\_qtss  
\_sandbox  
\_screensaver  
\_scsd  
\_securityagent  
\_serialnumberd  
\_softwareupdate  
\_spotlight  
\_sshd  
\_svn

```
_taskgated
_teamsserver
_timezone
_tokend
_trustevaluationagent
_unknown
_update_sharing
_usbmuxd
_uucp
_warmd
_webauthserver
_windowserver
_www
_wwwproxy
_xcsbuildagent
_xcscredserver
_xsverdocs
alessandrovivas
daemon
macports
nobody
root
```

---

Se quiser informações detalhadas sobre um usuário utilize o comando da Listagem 5.10.

#### Listagem 5.10: Informações Detalhadas

```
musashi:LivroMac-1aEdicao alessandrovivas$ dscl . -read /Users/alessandrovivas/
dsAttrTypeNative:_writers_hint: alessandrovivas
dsAttrTypeNative:_writers_jpegphoto: alessandrovivas
dsAttrTypeNative:_writers_passwd: alessandrovivas
dsAttrTypeNative:_writers_picture: alessandrovivas
dsAttrTypeNative:_writers_realname: alessandrovivas
dsAttrTypeNative:_writers_UserCertificate: alessandrovivas
dsAttrTypeNative:accountPolicyData:
<?xml version=      encoding=      ?>
<!DOCTYPE plist PUBLIC
          >
<plist version=      >
<dict>
```

---

## 5.8 Comando id

Quando se cria um usuário, **login**, no **Mac OS X**, ele recebe um identificador numérico (inteiro) indicando o número do usuário no sistema. É como se fosse o CPF do usuário no sistema e qualquer manipulação será realizada em cima do número do usuário e não do nome. Como todos sabem computadores são melhores em manipulação de números, índices, do que realizar operações em nomes.

O comando **id** faz a relação entre usuários e identificadores de usuários. A Listagem 5.11 apresenta o resultado do comando e como pode ser observado meu ID é 501.

#### Listagem 5.11: Identificadores no Mac

```
musashi:LivroMac-1aEdicao alessandrovivas$ id
uid=501(alessandrovivas) gid=20(staff) groups=20(staff),402(com.apple.sharepoint.group.1)
,502(access_bpf),12(everyone),61(localaccounts),79(_appserverusr),80(admin),81(
_appserveradm),98(_lpadmin),33(_appstore),100(_lpoperator),204(_developer),395(com.
apple.access_ftp),398(com.apple.access_screensharing),399(com.apple.access_ssh)
```

## 5.9 Alterando a Senha

O comando **passwd** permite a alteração da senha pelo usuário a qualquer tempo. A Listagem 5.12 apresenta o procedimento para alteração da senha.

### Listagem 5.12: Alterando a Senha

```
musashi:LivroMac-1aEdicao alessandrovivas$ passwd
Changing password for alessandrovivas.
Old Password:
```

Será necessário fornecer a senha atual para que o sistema autorize a mudança. A nova senha deve ser digitada e redigitada para que a mudança seja efetuada.

## 5.10 Usuários Logados

Quer descobrir quais são os usuários que estão logados nos sistema? Utilize o comando **users** da Listagem 5.13.

### Listagem 5.13: Usuários Logados

```
musashi:LivroMac-1aEdicao alessandrovivas$ users
alessandrovivas
```

## 5.11 Comandos de Calendário, Data e Hora

### 5.11.1 Comando Date

O comando **date** exibe a hora e data do sistema. A sintaxe do comando é igual para o Linux e o Mac. A Listagem 5.14 apresenta o resultado do comando **date**.

### Listagem 5.14: Visualizando Data e hora

```
musashi:LivroMac-1aEdicao alessandrovivas$ date
Qua 25 Nov 2015 15:42:00 BRST
```

### 5.11.2 Comando cal

Exibe um calendário em formato texto no terminal. Pode-se definir o ano e nesse caso todos os meses serão apresentados. Pode-se definir o mês específico. Se nenhum parâmetro for passado, o mês atual é exibido. A Listagem 5.15 apresenta o comando **cal**.

### Listagem 5.15: Comando cal

```
musashi:LivroMac-1aEdicao alessandrovivas$ cal
Novembro 2015
Do Se Te Qu Qu Se Sá
```

```

1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30

```

---

Se quiser imprimir o calendário de um ano inteiro utilize a Listagem 5.16.

#### Listagem 5.16: Imprimindo o Ano de 2030

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ cal 2030
```

A Figura 5.1 apresenta o resultado do comando da Listagem 5.16.

2030											
<b>Janeiro</b>				<b>Fevereiro</b>				<b>Março</b>			
Do	Se	Te	Qu	Qu	Se	Sá	Do	Se	Te	Qu	Qu
1	2	3	4	5	6	7	1	2	3	4	5
6	7	8	9	10	11	12	3	4	5	6	7
13	14	15	16	17	18	19	10	11	12	13	14
20	21	22	23	24	25	26	17	18	19	20	21
27	28	29	30	31			24	25	26	27	28
							29	30	31		
<b>Abril</b>				<b>Maio</b>				<b>Junho</b>			
Do	Se	Te	Qu	Qu	Se	Sá	Do	Se	Te	Qu	Qu
1	2	3	4	5	6	7	1	2	3	4	5
7	8	9	10	11	12	13	5	6	7	8	9
14	15	16	17	18	19	20	12	13	14	15	16
21	22	23	24	25	26	27	19	20	21	22	23
28	29	30					26	27	28	29	30
							31				
<b>Julho</b>				<b>Agosto</b>				<b>Setembro</b>			
Do	Se	Te	Qu	Qu	Se	Sá	Do	Se	Te	Qu	Qu
1	2	3	4	5	6	7	1	2	3	4	5
7	8	9	10	11	12	13	4	5	6	7	8
14	15	16	17	18	19	20	11	12	13	14	15
21	22	23	24	25	26	27	18	19	20	21	22
28	29	30	31				25	26	27	28	29
							31				
<b>Outubro</b>				<b>Novembro</b>				<b>Dezembro</b>			
Do	Se	Te	Qu	Qu	Se	Sá	Do	Se	Te	Qu	Qu
1	2	3	4	5	6	7	1	2	3	4	5
6	7	8	9	10	11	12	3	4	5	6	7
13	14	15	16	17	18	19	10	11	12	13	14
20	21	22	23	24	25	26	17	18	19	20	21
27	28	29	30	31			24	25	26	27	28
							30	31			

Figura 5.1: Calendário de 2030

Se quiser imprimir um mês de um ano específico utilize a Listagem 5.17.

#### Listagem 5.17: Imprime Mês e Ano

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ cal 11 1973
```

Novembro 1973

```

Do Se Te Qu Qu Se Sá
      1   2   3
    4   5   6   7   8   9   10
  11  12  13  14  15  16  17
  18  19  20  21  22  23  24
  25  26  27  28  29  30

```

Para mostrar o dias do ano a Listagem 5.18.

#### Listagem 5.18: Domingo como Primeiro Dia da Semana

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ cal -j
```

Dezembro 2015

```

Do Se Te Qu Qu Se Sá
      335 336 337 338 339
    340 341 342 343 344 345 346
    347 348 349 350 351 352 353
    354 355 356 357 358 359 360

```

### 5.11.3 Comando calendar

O comando **calendar** lê seu arquivo de calendário e imprime os eventos relacionados a uma data específica.

## 5.12 Comando chflags

O comando **chflags** pode ser utilizada para tornar oculto um arquivo. Quando oculto ele não será visualizado no **Finder**. A Listagem 5.19 mostra os atributos do arquivo cidades.txt.

#### Listagem 5.19: Atributos do Arquivo cidades.txt

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ ls -lF cidades.txt
-rw-r--r-- 1 alessandrovivas staff 89672 25 Nov 11:13 cidades.txt
```

---

Na Listagem 5.20 é alterado o estado do arquivo para oculto.

#### Listagem 5.20: Alterando o Estado do Arquivo

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ chflags hidden cidades.txt
```

---

Na Listagem 5.21 são listados os atributos do arquivo e o aparece o símbolo @ que indica que o arquivo está oculto.

#### Listagem 5.21: Atributos do Arquivo cidades.txt

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ ls -lF cidades.txt
-rw-r--r--@ 1 alessandrovivas staff 89672 25 Nov 11:13 cidades.txt
```

---

Para tornar o arquivo visível novamente utilize o comando da Listagem 5.22.

#### Listagem 5.22: Alterando os Atributos do Arquivo para Visível

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ chflags nohidden cidades.txt
```

---

## 5.13 Comando finger

O comando **finger** fornece informações sobre os usuários cadastradas no sistema. Dentre essas informações estão: nome, login, diretório inicial, último login efetuado com sucesso, shell de uso. Ao se executar o comando **finger** sem argumentos, ele exibe num formato padrão de informação como na Listagem 5.23. A coluna Login é o nome do login do usuário, a coluna Name é o nome completo do usuário, Tty é o terminal onde o usuário está logado, Idle mostra o tempo ocioso, Login Time mostra a data e a hora quando o usuário logou, Office mostra a localização física do usuário e Phone mostra o telefone do usuário

#### Listagem 5.23: Comando finger

```
musashi:LivroMac-1aEdicao alessandrovivas$ finger
Login      Name          TTY  Idle  Login Time  Office  Phone
alessandrovivas Alessandro Vivas  *con   1d  Seg    22:12
alessandrovivas Alessandro Vivas     s00        Ter    06:43
```

---

Utilize o argumento `-l` para listar informações de todos os usuários logados na máquina naquele momento como na Listagem 5.24.

#### Listagem 5.24: Comando finger -l

```
musashi:LivroMac-1aEdicao alessandrovivas$ finger -l
Login: alessandrovivas      Name: Alessandro Vivas
Directory: /Users/alessandrovivas    Shell: /bin/bash
On since Seg 23 Nov 22:12 (BRST) on console, idle 1 day 17:45 (messages off)
On since Ter 24 Nov 06:43 (BRST) on ttys000
No Mail.
No Plan.
```

---

Se quiser obter informações sobre um usuário específico utilize o comando da Listagem 5.25.

#### Listagem 5.25: Comando finger no Linux

```
musashi:LivroMac-1aEdicao alessandrovivas$ finger alessandrovivas
Login: alessandrovivas      Name: Alessandro Vivas
Directory: /Users/alessandrovivas    Shell: /bin/bash
On since Seg 23 Nov 22:12 (BRST) on console, idle 1 day 17:52 (messages off)
On since Ter 24 Nov 06:43 (BRST) on ttys000
On since Qua 25 Nov 15:40 (BRST) on ttys001 (messages off)
No Mail.
No Plan.
```

## 5.14 Comando last

O comando **last** imprime informações do último **login** de um determinado usuário. A Listagem 5.26 apresenta o resultado e sintaxe do comando.

#### Listagem 5.26: Comando last

```
musashi:LivroMac-1aEdicao alessandrovivas$ last alessandrovivas
wtmp begins Wed Nov 25 15:40
```

## 5.15 Comando vm\_stat

O comando **vm\_stat** mostra a estatística de uso de memória virtual, incluindo memória livre total, memória utilizada, memória física, memória swap, memória compartilhada e buffers utilizados pelo kernel. A Listagem 5.27 apresenta o resultado do comando.

#### Listagem 5.27: Comando free

```
Mach Virtual Memory Statistics: (page size of 4096 bytes)
Pages free:                      2151369.
Pages active:                     796598.
Pages inactive:                   127372.
Pages speculative:                 411792.
Pages throttled:                  0.
Pages wired down:                 706471.
Pages purgeable:                  72490.
```

```
:          3674787.  
Pages copy-on-write:      147901.  
Pages zero filled:       2462641.  
Pages reactivated:        10.  
Pages purged:            0.  
File-backed pages:       620935.  
Anonymous pages:         714827.  
Pages stored in compressor: 0.  
Pages occupied by compressor: 0.  
Decompressions:          0.  
Compressions:            0.  
Pageins:                 151389.  
Pageouts:                0.  
Swapins:                 0.  
Swapouts:                0.
```

---

## 5.16 Comando su

Executa o interpretador de comandos com a substituição do usuário e do grupo. Possibilidade de logar imediatamente no mesmo terminal em uso com outro usuário. Prática comum de super-usuário. A Listagem 5.28 apresenta a execução do comando **su**.

### Listagem 5.28: Logar como Super Usuário

```
$ su -  
Password:
```

---

## 5.17 Comando uname

O comando **uname** é utilizado para apresentar informações sobre o sistema operacional de sua máquina. A Listagem 5.29 apresenta o comando, para verificar qual sistema operacional está utilizando.

### Listagem 5.29: Verificar Informações sobre o Mac

```
musashi:LivroMac-1aEdicao alessandrovivas$ uname -s  
Darwin
```

---

Para verificar a versão do seu kernel utilize o comando apresentando na Listagem 5.30.

### Listagem 5.30: Verificar sua Versão do kernel

```
musashi:LivroMac-1aEdicao alessandrovivas$ uname -r  
15.0.0
```

---

A Listagem 5.31 apresenta o comando para verificar se sua plataforma é de 32 ou 64 bits. Neste caso a plataforma é de 64 bits, pois a resposta foi **x86\_64**.

### Listagem 5.31: Verificando a Plataforma

```
musashi:LivroMac-1aEdicao alessandrovivas$ uname -m  
x86_64
```

---

Para descobrir o nome de sua máquina utilize o comando **uname -n**. A Listagem 5.32 apresenta o resultado.

#### Listagem 5.32: Verificar o Nome de sua Máquina

```
musashi:LivroMac-1aEdicao alessandrovivas$ uname -n  
musashi-2.local
```

Para apresentar todas as informações sobre seu sistema operacional utilize o comando **uname -a**. A Listagem 5.33 apresenta o resultado.

#### Listagem 5.33: Apresenta todas as informações sobre seu sistema operacional

```
musashi:LivroMac-1aEdicao alessandrovivas$ uname -a  
Darwin musashi-2.local 15.0.0 Darwin Kernel Version 15.0.0: Sat Sep 19 15:53:46 PDT 2015;  
root:xnu-3247.10.11~1/RELEASE_X86_64 x86_64
```

## 5.18 Comando **uptime**

O comando **uptime** apresenta as seguintes informações: a hora corrente, há quanto tempo o seu computador está ligado, quantidade de usuários logados e a carga média do sistema a 1, 5 e 15 minutos passados. A Listagem 5.34 ilustra o resultado do comando.

#### Listagem 5.34: Tempo de Funcionamento

```
musashi:LivroMac-1aEdicao alessandrovivas$ uptime  
16:31 up 1 day, 18:20, 2 users, load averages: 3,38 3,63 3,59
```

## 5.19 Variável PATH

Ao se digitar um comando, o arquivo binário deve ser localizado pelo sistema operacional para ser executado. Se ele não o encontra uma mensagem de erro é exibida em seguida. Algumas vezes, quando criamos, por exemplo, arquivos executáveis, necessitamos passar o local do arquivo como ilustrado na Listagem 5.35.

#### Listagem 5.35: Localização de um Comando

```
./arquivo_executavel  
./usr/bin/arquivo_executavel
```

Acima, o ponto indica o caminho desde o diretório raiz até o diretório corrente. Entretanto, existem locais padrões a serem buscados e tais locais são definidos por uma variável chamada **PATH**.

Quando você digita um comando e o shell não encontra, pode estar acontecendo duas coisas: o comando não foi instalado ou o seu shell não está procurando no local correto. Para saber todos os caminhos onde seu shell use a Listagem 5.36.

#### Listagem 5.36: Variável PATH

```
musashi:LivroMac-1aEdicao alessandrovivas$ echo $PATH  
/Library/Frameworks/Python.framework/Versions/3.4/bin:/usr/local/bin:/usr/bin:/bin:/usr/  
sbin:/sbin:/opt/X11/bin:/Library/TeX/texbin
```

Os caminhos são separados por dois-pontos (:). No exemplo acima o primeiro local onde o shell procura os comandos é o diretório `/Library/Frameworks/Python.framework/Versions/3.4/bin`. Como pode ser observado, o shell não procura diretamente em seu diretório. Você pode imaginar o motivo? Razões de segurança: se um intruso inserisse um programa malicioso(ex. uma modificação do comando `ls`) e se o shell procurasse diretamente no seu diretório, a execução desse programa poderia danificar seus arquivos.

## 5.20 Comando w

O comando **w** verifica quais usuários estão logados, que eles estão fazendo e quanto tempo estão logados. A Listagem 5.37 apresenta o resultado do comando.

### Listagem 5.37: Comando w

```
musashi:LivroMac-1aEdicao alessandrovivas$ w
16:35 up 1 day, 18:24, 2 users, load averages: 3,51 3,65 3,59
USER      TTY      FROM          LOGIN@    IDLE WHAT
alessandrovivas  console   -           Seg22    1day   -
alessandrovivas s000      -           Ter06     - w
```

## 5.21 Comando whereis

O comando **whereis** determina a localização de seu programa executável(binário), fonte e páginas de manual referente a um comando. Ele é mais completo que o comando **which**.

Se quiser encontrar a localização de um programa, por exemplo o `g++`, utilizamos o exemplo na Listagem 5.38.

### Listagem 5.38: Comando whereis

```
musashi:LivroMac-1aEdicao alessandrovivas$ whereis g++
/usr/bin/g++
```

## 5.22 Comando which

O comando **which** procura somente no **PATH** o comando desejado. Se quiser encontrar a localização de um programa, por exemplo o `g++`, utilizamos o exemplo na Listagem ??.

### Listagem 5.39: Comando which

```
musashi:LivroMac-1aEdicao alessandrovivas$ which g++
/usr/bin/g++
```

## 5.23 Procurando Informações

### 5.23.1 Comando locate

O comando **locate** lista arquivos que contenham o texto dado (semelhante ao whereis). A Listagem 5.40 utiliza o comando para listar todas as ocorrências de **fdisk**.

#### Listagem 5.40: Comando locate

```
musashi:LivroMac-1aEdicao alessandrovivas$ locate fdisk
/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/
    MacOSX10.11.sdk/usr/share/man/man8/fdisk.8
/usr/local/Library/Formulas/gptfdisk.rb
/usr/sbin/fdisk
/usr/share/man/man8/fdisk.8
```

---

### 5.23.2 Comando find

O comando **find** lista arquivos de acordo com o padrão enviado a ele. A Listagem 5.41 utiliza o comando para listar todas as ocorrências de **Mac\*.pdf**.

#### Listagem 5.41: Comando find

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ find Mac*.pdf
Mac1aEdicao.pdf
```

---

### 5.23.3 Comando whatis

O comando **whatis** pode ser utilizado para busca no banco de dados do sistema sobre informações de comandos como na Listagem 5.42.

#### Listagem 5.42: Comando whatis

```
vivas@zafu:~$ whatis clear
clear (1)           - clear the terminal screen
vivas@zafu:~$ whatis ps
ps (1)             - report a snapshot of the current processes.
vivas@zafu:~$ whatis netstat
netstat (8)         - Mostra conexões de rede, tabelas de roteamento, estatísticas de
                      interface e conexões...
vivas@zafu:~$ whatis route
route (8)           - mostra / manipula a tabela de roteamento IP
vivas@zafu:~$ whatis ls
ls (1)              - list directory contents
LS (6)              - display animations aimed to correct users who accidentally enter
                      LS instead of ls .
vivas@zafu:~$ whatis touch
touch (1)            - change file timestamps
vivas@zafu:~$ whatis cat
cat (1)              - concatenate files and print on the standard output
vivas@zafu:~$ whatis date
date (1)             - print or set the system date and time
```

---

## 5.24 Rodando Comandos

### 5.24.1 Rodando múltiplos comandos

Para rodar múltiplos comandos basta separar com ponto e vírgula. A Listagem 5.43 apresenta o resultado do comando.

#### Listagem 5.43: Rodando Múltiplos Comandos

```
musashi:LivroMac-1aEdicao alessandrovivas$ hostname; date; time;
musashi-2.local
3Qua 25 Nov 2015 19:47:41 BRST

real 0m0.000s
user 0m0.000s
sys 0m0.000s
```

O exemplo ilustrado na Listagem 5.43 mostra como rodar comandos em sequência, de forma que eles sempre serão executados, independente do resultado da execução do comando anterior. Uma outra forma é condicionar a execução do comando subsequente à execução do primeiro comando. Podemos utilizar o `&&` para executar o segundo comando apenas se o primeiro comando tiver sucesso. Outra opção é condicionar a execução do segundo ao fracasso do primeiro comando, ou seja, o segundo só será executado se o primeiro retornar erro. Para este fim utilizaremos o `||`. Ambos exemplos são ilustrados na Listagem 5.44.

#### Listagem 5.44: Rodando Múltiplos Comandos Condicionados

```
$ test -f teste.existe && echo 'existe'
$ test -f teste.existe || echo 'existe'
$ touch teste.existe && test -f teste.existe && echo 'existe'
```

Note que o primeiro só irá imprimir ‘existe’ na tela se o arquivo `teste.existe` já existir no diretório corrente. Caso o arquivo não exista, apenas a terceira linha imprimirá ‘existe’ na tela, umas vez que o arquivo acaba de ser criado pelo comando `touch`.

### 5.24.2 Rodando um comando em background

Outra forma muito útil de executar programas é executá-los em background. Isto implica que o programa estará rodando, mas você não verá os resultados e não poderá passar sinais ao programa através do teclado. Isto implicará que o programa não poderá ser finalizado utilizando a combinação de teclas `<Ctrl-C>`. Ao executar um programa em background, o shell ficará livre para a execução de outros programas. Para executar um programa em background, você deverá utilizar o `&` após chamar o programa. Veja o exemplo na Listagem 5.45, no qual o `emacs` será executado em background.

#### Listagem 5.45: Rodando Comando em Background

```
$ emacs &
```



# Gerenciamento de Processos

*Os americanos precisam do telefone,  
nós não. Nós temos muitos garotos  
mensageiros.*

Sir William Preece, chefe da agência  
britânica de correios em 1876

## Sumário

6.1	Rodando Comandos em Background com bg e em Foreground fg . . . . .	58
6.2	Visualizando Todos os Processos em Execução . . . . .	59
6.3	Todos os Processos de um Usuário Específico . . . . .	59
6.4	Lista de Processos Ordenadas pelo Consumo de CPU . . . . .	60
6.5	Lista dos Processos que Mais Consomem Memória . . . . .	60
6.6	Obtendo Informações de um Processo Específico . . . . .	61
6.7	Comando top . . . . .	61
6.8	Listando todos os Sinais com o Comando kill . . . . .	62
6.9	Matando um Processo com o Comando Kill . . . . .	62
6.10	Comando killall . . . . .	62
6.11	Comando time . . . . .	63

## 6.1 Rodando Comandos em Background com **bg** e em Foreground **fg**

O comando **bg** é usado para rodar programas em background. Vamos supor que você esteja trabalhando em um terminal e necessita listar todos os arquivos de seu computador e colocar em um arquivo txt. Normalmente você deveria ficar esperando o final do resultado do comando pois o terminal estaria ocupado. Utilize o símbolo & para rodar o comando em background. O resultado do comando será o número do **Process ID (PID)** do processo como mostrado na Listagem 6.1.

### Listagem 6.1: Rodar Comandos em Background

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ ls -R / > teste.txt &
[2] 23043
```

Para colocar o programa rodando no terminal novamente utilize o comando **fg** utilizando o PID do processo como na Listagem 6.2. O primeiro passo é rodar o comando **jobs** para descobrir o identificador dele. Se ele foi o primeiro comando a ser executado em background seu identificador será 1. Na Listagem 6.1 o identificador é 2.

### Listagem 6.2: Rodando em Foreground

```
musashi-2:processamega alessandrovivas$ ls -R / > teste.txt &
[1] 23317
musashi-2:processamega alessandrovivas$ jobs -l
[1]+ 23317 Running                  ls -R / > teste.txt &
musashi-2:processamega alessandrovivas$ fg 1
ls -R / > teste.txt
```

Para rodar um programa em background com **bg** vamos colocar um comando para rodar como na Listagem 6.3.

### Listagem 6.3: Uso do bg

```
musashi-2:processamega alessandrovivas$ ping www.ufsj.edu.br
PING www.ufsj.edu.br (200.17.67.239): 56 data bytes
64 bytes from 200.17.67.239: icmp_seq=0 ttl=45 time=41.994 ms
```

...

Abra outro terminal e descubra o PID do processo com o comando da Listagem 6.4.

### Listagem 6.4: Descubra o PID

```
musashi-2:processamega alessandrovivas$ ps -aef | grep ping
501 23365 23309  0  7:34    ttys000    0:00.01 ping www.ufsj.edu.br
```

O PID do processo é 23365 e agora abra outro terminal e pare a execução do processo ping como na Listagem 6.5.

### Listagem 6.5: Pare a Execução do Processo

```
musashi-2:processamega alessandrovivas$ kill -s SIGSTOP 23365
```

Ao abrir o terminal onde foi digitado o comando você verificará o resultado onde o número 1 representa o número do job, Listagem 6.6.

#### Listagem 6.6: Parando a Execução do Comando

```
[1]+ Stopped ping www.ufsj.edu.br
```

Se quiser rodar o processo novamente abra o terminal onde ele estava rodando e use o código da Listagem 6.7.

#### Listagem 6.7: Executando novamente o processo

```
musashi-2:processamega alessandrovivas$ bg 1
[1]+ ping www.ufsj.edu.br &
musashi-2:processamega alessandrovivas$ 64 bytes from 200.17.67.239: icmp_seq=61 ttl=45
time=42.964 ms
64 bytes from 200.17.67.239: icmp_seq=62 ttl=45 time=40.868 ms
```

## 6.2 Visualizando Todos os Processos em Execução

Definimos processo como um programa em execução. Podemos definir como espaço de endereçamento de um processo como: código do programa, variáveis utilizadas, pilha do processo e outras informações necessárias.

Os processos são constituídos da localização do espaço de endereçamento, status, prioridade de execução, informações sobre os recursos utilizados, máscara de sinal do processo e identificação do proprietário. Os principais atributos são: identificador do processo (PID), Identificador do processo pai (PPID), prioridade de execução (nice), TTY (terminal), identificação real do usuário e do grupo.

O comando **ps** com a opção **-aef** apresenta todos os processos em execução como na Listagem 6.8.

#### Listagem 6.8: Todos os Processos em Execução

```
$ ps -aef
musashi:$ ps -aef
  UID  PID  PPID   C STIME   TTY         TIME CMD
    0    1      0   0 Seg10   ??        10:26.13 /sbin/launchd
    0    46     1   0 Seg10   ??        3:27.51 /usr/libexec/UserEventAgent (System)
    0    47     1   0 Seg10   ??        3:29.43 /usr/sbin/syslogd
    0    49     1   0 Seg10   ??        0:03.01 /usr/libexec/kextd
    0    50     1   0 Seg10   ??        2:54.95 /System/Library/Frameworks/CoreServices.
                                         framework/Versions/A/Frameworks/FSEvents.framework/Versions/A/Support/fsevents
  55    54     1   0 Seg10   ??        0:02.73 /System/Library/CoreServices/
                                         appleeventsd --server
    0    55     1   0 Seg10   ??        1:21.96 /usr/libexec/configd
    0    56     1   0 Seg10   ??        0:48.23 /System/Library/CoreServices/powerd.
                                         bundle/powerd
    0    62     1   0 Seg10   ??        1:06.00 /usr/libexec/airportd
    0    64     1   0 Seg10   ??        0:00.80 /usr/libexec/warmd
    0    65     1   0 Seg10   ??        7:04.01 /System/Library/Frameworks/CoreServices.
                                         framework/Frameworks/Metadata.framework/Support/mds
...
```

## 6.3 Todos os Processos de um Usuário Específico

O comando **ps** com a opção **-u** pode ser utilizado para visualizar todos os processos de um determinado usuário como na Listagem 6.9.

#### Listagem 6.9: Todos os Processos em Execução de um Usuário Específico

```
musashi:$ ps -u alessandrovivas
UID  PID  TTY      TIME CMD
501  298  ??      0:10.20 /usr/libexec/UserEventAgent (Aqua)
501  300  ??      18:08.42 /usr/sbin/distnoted agent
501  301  ??      0:07.37 /usr/sbin/universalaccessd launchd -s
501  302  ??      0:28.04 /usr/sbin/cfprefsd agent
501  303  ??      0:02.62 /usr/libexec/lsd
501  306  ??      1:35.98 /System/Library/CoreServices/Dock.app/Contents/MacOS/Dock
501  309  ??      5:02.25 /System/Library/CoreServices/SystemUIServer.app/Contents/
MacOS/SystemUIServer
501  310  ??      19:05.71 /System/Library/CoreServices/Finder.app/Contents/MacOS/
Finder
501  313  ??      0:00.02 /usr/sbin/pboard
501  318  ??      1:36.89 /System/Library/Frameworks/ApplicationServices.framework/
Frameworks/ATS.framework/S
501  319  ??      0:04.10 /System/Library/CoreServices/sharedfilelistd
501  322  ??      0:02.76 /usr/libexec/sharingd
501  323  ??      0:24.66 /usr/libexec/pkd
501  324  ??      0:04.64 /System/Library/PrivateFrameworks/TCC.framework/Resources/
tccd
501  325  ??      0:07.67 /usr/sbin/usernoted
...
...
```

## 6.4 Lista de Processos Ordenadas pelo Consumo de CPU

Para obter uma lista de processos ordenadas pelo consumo de CPU utilizamos a Listagem 6.10. Neste caso utilizamos o comando **head** para imprimir somente as 5 primeiros ocorrencias.

#### Listagem 6.10: Processos que Mais Consomem CPU

```
musashi:LivroMac-1aEdicao alessandrovivas$ ps -aef -r | head -5
UID  PID  PPID  C STIME   TTY      TIME CMD
501  418     1   0 Seg10    ??      1882:46.68 /Applications/Dropbox.app/Contents/
MacOS/Dropbox
501 29759     1   0  9:32    ??      12:22.55 /System/Library/Frameworks/MediaLibrary.
framework/Versions/A/XPCServices/com.apple.MediaLibraryService.xpc/Contents/MacOS/
com.apple.MediaLibraryService
501 29762     1   0  9:32    ??      2:23.05 /Library/Frameworks/iTunesLibrary.
framework/Versions/A/XPCServices/com.apple.iTunesLibraryService.xpc/Contents/MacOS/
com.apple.iTunesLibraryService
501 10370     1   0 Ter06    ??      13:40.44 /Applications/Utilities/Terminal.app/
Contents/MacOS/Terminal
```

## 6.5 Lista dos Processos que Mais Consomem Memória

Para obter uma lista de processos ordenadas pelo consumo de memória utilizamos a Listagem 6.11. Neste caso utilizamos o comando **head** para imprimir somente as 5 primeiros ocorrencias.

#### Listagem 6.11: Processos que Mais Consomem CPU

```
musashi:LivroMac-1aEdicao alessandrovivass$ ps -aef -m | head -5
UID  PID  PPID  C STIME   TTY      TIME CMD
501  418     1  0 Seg10  ??        1889:11.25 /Applications/Dropbox.app/Contents/
    MacOS/Dropbox
501 1152     1  0 Seg10  ??        71:54.59 /System/Library/Frameworks/WebKit.
    framework/Versions/A/XPCServices/com.apple.WebKit.WebContent.xpc/Contents/MacOS/com
    .apple.WebKit.WebContent
501 1594     1  0 Seg10  ??        134:02.82 /System/Library/Frameworks/WebKit.
    framework/Versions/A/XPCServices/com.apple.WebKit.WebContent.xpc/Contents/MacOS/com
    .apple.WebKit.WebContent
501 1258     1  0 Seg10  ??        4:09.52 /System/Library/Frameworks/WebKit.
    framework/Versions/A/XPCServices/com.apple.WebKit.WebContent.xpc/Contents/MacOS/com
    .apple.WebKit.WebContent
```

## 6.6 Obtendo Informações de um Processo Específico

Para visualizar informações de um processo específico utilizamos o comando **ps** em conjunto com o comando **grep** como na Listagem 6.12.

#### Listagem 6.12: Obtendo Informações de um Processo Específico

```
$ ps -aef | grep texmaker
 501 61850     1  0  4:05  ??        1:28.48 /Applications/texmaker.app/Contents/
    MacOS/texmaker
 501 62417 35019  0  4:27  ttys001    0:00.00 grep texmaker
```

## 6.7 Comando top

O comando **top** é utilizado para obter informações sobre os processos que estão rodando em sua máquina. A Listagem 6.13 apresenta do resultado do comando.

Os estados possíveis de um processo são:

- runnable - rodando
- sleeping - está esperando por um evento
- swapped - não está executando e foi armazenado na memória virtual
- zombie - está tentando morrer (pode ter perdido seu pai)
- stopped - está proibido de executar (através de CTRL-Z ou um SIGSTOP)

#### Listagem 6.13: Comando top

```
Processes: 330 total, 3 running, 9 stuck, 318 sleeping, 2558 threads
Load Avg: 6.08, 4.74, 4.00 CPU usage: 33.64% user, 19.66% sys, 46.68% idle
SharedLibs: 180M resident, 20M data, 27M linkedit.
MemRegions: 97078 total, 7892M resident, 173M private, 1016M shared. PhysMem: 16G used (1984M wired), 171M unused.
VM: 886G vsize, 529M framework vsize, 5879(0) swapins, 12473(0) swapouts.
Networks: packets: 3403992/2568M in, 3131219/979M out. Disks: 2931560/175G read, 4130346/96G written.

PID  COMMAND  %CPU  TIME  #TH  #WQ  #PORT  MEM  PURG  CMPRS  PGRP  PPID  STATE  BOOSTS
56882  top      4.7  00:01.45 1/1  0    21   4156K  0B  0B    56882 10373  running  *0[1]
56879  mdworker  0.0  00:00.04 3   0    43   1636K  0B  0B    56879 1       sleeping *0[1]
56778  com.apple.au 0.0  00:00.06 2   1    35   1604K  0B  0B    56778 1       sleeping 0[1]
56777  com.apple.au 0.0  00:00.01 2   1    22   976K   0B  0B    56777 1       sleeping 0[2]
56771  com.apple.We 1.3  00:29.28 18  5    262  210M+ 19M  0B  56771 1       sleeping *0[1717+]
56695- Microsoft AU 0.0  00:00.13 2   0    97   4052K  0B  0B    56695 1       sleeping *0[5]
56691- Microsoft Ex 0.1  01:18.95 9   2    189  284M  300K  0B  56691 1       sleeping *0[85]
```

```

56669 com.apple.iC 0.0 00:00.37 2 0 50 2132K 0B 0B 56669 1 sleeping 0[1]
56668 syncdefaults 0.0 00:08.28 4 0 106- 7868K- 0B 0B 56668 1 sleeping 0[18]
56609 CoreServices 0.0 00:00.16 3 0 138- 4232K- 0B 0B 56609 1 sleeping *0[1]
56608- AdobeReader 0.1 00:10.50 16 3 234- 61M- 0B 0B 56608 1 sleeping *0[73]
56557 mdworker 0.0 00:00.04 3 0 47 1672K 0B 0B 56557 1 sleeping *0[1]
56553 mdworker 0.0 00:00.07 3 0 59 1876K 0B 0B 56553 1 sleeping *0[1]
56549 mdworker 0.0 00:00.06 3 0 49 1776K 0B 0B 56549 1 sleeping *0[1]
56548 mdworker 0.0 00:00.05 3 0 49 1732K 0B 0B 56548 1 sleeping *0[1]
56547 mdworker 0.0 00:00.05 3 0 49 1736K 0B 0B 56547 1 sleeping *0[1]
56546 mdworker 0.0 00:00.05 3 0 49 1752K 0B 0B 56546 1 sleeping *0[1]
56450 com.apple.We 0.0 00:04.95 12 2 240 63M 80K 0B 56450 1 sleeping *0[1639]
56162 com.apple.We 0.0 00:07.52 12 2 233 47M+ 236K 0B 56162 1 sleeping *0[2321]
56077 ocsdp 0.0 00:00.19 4 0 68- 2428K- 0B 0B 56077 1 sleeping *0[1]
55786 com.apple.We 0.1 00:19.85 15 2 241 211M+ 260K 0B 55786 1 sleeping *0[5959+]
55632 com.apple.We 0.0 00:50.39 14 2 197 132M 344K 0B 55632 1 sleeping *0[13067]
51556- dbfsevents 0.0 00:11.82 1 0 7 192K 0B 0B 418 51555 sleeping *0[1]
...

```

---

## 6.8 Listando todos os Sinais com o Comando kill

O comando **kill** é utilizado para enviar sinais para um processo. A Listagem 6.14 apresenta todos os sinais disponíveis.

**Listagem 6.14:** Comando kill

```

musashi:LivroMac-1aEdicao alessandrovivas$ kill -l
 1) SIGHUP  2) SIGINT   3) SIGQUIT  4) SIGILL
 5) SIGTRAP  6) SIGABRT  7) SIGEMT   8) SIGFPE
 9) SIGKILL 10) SIGBUS   11) SIGSEGV 12) SIGSYS
13) SIGPIPE 14) SIGALRM 15) SIGTERM  16) SIGURG
17) SIGSTOP 18) SIGTSTP 19) SIGCONT  20) SIGCHLD
21) SIGTTIN 22) SIGTTOU 23) SIGIO   24) SIGXCPU
25) SIGXFSZ 26) SIGVTALRM 27) SIGPROF 28) SIGWINCH
29) SIGINFO 30) SIGUSR1 31) SIGUSR2

```

---

## 6.9 Matando um Processo com o Comando Kill

Para eliminar um processo com o comando **kill** precisamos saber o PID do processo. Vamos supor que você deseja eliminar o processo do firefox. O primeiro passo é saber o PID do firefox e para isto faça o procedimento da Listagem 6.15.

**Listagem 6.15:** Comando kill

```

$ ps -aef | grep firefox
501 79667 1 0 7:52 ?? 19:08.19 /Applications/Firefox.app/Contents/MacOS
/firefox -foreground

```

---

Como podemos perceber o PID do firefox é 79667. Agora para matar o processo firefox enviamos um sinal SIGKILL, número 9, como na Listagem 6.16.

**Listagem 6.16:** Matando o Processo Firefox

```
$ kill -9 79667
```

---

## 6.10 Comando killall

O comando **killall** permite matar um processo pelo nome e não pelo **PID**. A Listagem 6.17 roda o comando **ps** e verifica se o processo **texmaker** está rodando.

#### Listagem 6.17: Utilizando killall

```
% verifica se o processo texmaker esta rodando
musashi:LivroMac-1aEdicao alessandrovivas$ ps -ax | grep texmaker
10366 ?? 41:01.15 /Applications/texmaker.app/Contents/MacOS/texmaker
57182 ttys000 0:00.00 grep texmaker

% mata o processo texmaker
musashi:LivroMac-1aEdicao alessandrovivas$ killall texmaker

% mostra que o processo texmaker foi eliminado
musashi:LivroMac-1aEdicao alessandrovivas$ ps -ax | grep texmaker
57188 ttys000 0:00.00 grep texmaker
```

## 6.11 Comando time

O comando **time** é utilizado para mostrar o tempo de execução de um script ou processo. Este comando calcula o tempo utilizado pelo programa no modo usuário, no modo kernel e a quantidade de tempo realmente utilizado. A quantidade de tempo total utilizado é sempre maior do que a soma do tempo no modo Kernel mais o tempo no modo usuário. Ele envolve todos os tempos: processamento de interrupção e tempo de espera na fila de processos prontos ou aguardando o processamento de entrada e saída. A Listagem 6.18 calcula o tempo gasto pelo comando **date**. Neste caso ele gastou 2 ms de tempo no modo kernel, 1 ms no modo usuário e o tempo total de 4 ms.

#### Listagem 6.18: Calculando Tempo de Execução de um Programa ou Script

```
time date
Sex 4 Set 2015 13:54:02 BRT

real 0m0.004s
user 0m0.001s
sys 0m0.002s
```

Vamos o usar o comando da Listagem 6.18 em um notebook rodando Fedora. Os dois tem processadores semelhantes, mas o primeiro roda Mac OSX e o segundo Fedora. O resultado do comando é apresentado na Listagem 6.19. Como pode ser visualizado o notebook rodando Fedora teve um desempenho melhor do que o Mac OSX.

#### Listagem 6.19: Calculando Tempo de Execução de um Programa ou Script

```
time date
Sex Set 4 14:13:13 BRT 2015

real 0m0.002s
user 0m0.000s
sys 0m0.002s
```



# Permissão e Propriedade

*A Apple já está morta.*

Nathan Myhrvold, então diretor de tecnologia da Microsoft, em 1997

## Sumário

7.1 Comando chown . . . . .	66
7.2 Comando chgrp . . . . .	66
7.3 Permissão e Propriedade . . . . .	66
7.4 Comando chroot . . . . .	68

## 7.1 Comando chown

O comando **chown** é responsável por alterar o dono de um arquivo. Para verificar o proprietário de um arquivo podemos utilizar o comando **ls** como na Listagem 7.6.

### Listagem 7.1: Listando o Proprietário e as Permissões dos Arquivos

```
musashi:LivroMac-1aEdicao alessandrovivas$ ls -l cidades.txt
-rw-r--r-- 1 alessandrovivas staff 89672 25 Nov 11:13 cidades.txt
```

Percebemos que este arquivo pertence ao usuário alessandrovivas e está associado ao grupo staff. Se quiser alterar o proprietário deste arquivo para o usuário **nobody** utilize o comando da Listagem 7.2

### Listagem 7.2: Alterando o Proprietário do Arquivo

```
musashi:LivroMac-1aEdicao alessandrovivas$ sudo chown nobody cidades.txt
Password:
3musashi:LivroMac-1aEdicao alessandrovivas$ ls -l cidades.txt
-rw-r--r-- 1 nobody staff 89672 25 Nov 11:13 cidades.txt
```

## 7.2 Comando chgrp

O comando **chgrp** é responsável por alterar o grupo de um arquivo. Para verificar o grupo de um arquivo podemos utilizar o comando **ls** como na Listagem 7.3.

### Listagem 7.3: Verificando o Grupo

```
musashi:LivroMac-1aEdicao alessandrovivas$ ls -l cidades.txt
-rw-r--r-- 1 nobody staff 89672 25 Nov 11:13 cidades.txt
```

Percebemos que este arquivo pertence ao grupo staff. Se quiser alterar o grupo deste arquivo para o grupo **everyone** utilize o comando da Listagem 7.4

### Listagem 7.4: Alterando o Grupo do Arquivo

```
musashi:LivroMac-1aEdicao alessandrovivas$ sudo chgrp everyone cidades.txt
musashi:LivroMac-1aEdicao alessandrovivas$ ls -l cidades.txt
3-rw-r--r-- 1 nobody everyone 89672 25 Nov 11:13 cidades.txt
```

## 7.3 Permissão e Propriedade

Os sistemas UNIX supõem que a máquina pode ser compartilhada por diferentes usuários, por este motivo é necessário atribuir posse e permissões diferentes para cada usuário do sistema. Cada arquivo possui então um dono e permissões diferenciadas para quem é o dono e quem não é. Para verificar estas características em um arquivo, você pode usar o comando **ls** conforme o exemplo na Listagem 7.5.

### Listagem 7.5: Listando o Proprietário e as Permissões dos Arquivos

```
$ ls -l
-rw-rw-r-- 1 bob users 375600 Dez 5 14:30 myfile
```

O arquivo que foi listado no exemplo na Listagem 7.5 pertence ao usuário *bob*. Este, por sua vez, está no grupo de usuários chamado *users*. As permissões do arquivo estão listadas no código `-rw-rw-r-`. O primeiro traço à esquerda significa que este é um arquivo normal contendo qualquer tipo de dados. Um diretório teria um `d` ao invés do traço `-`. Os próximos 9 caracteres são as permissões do arquivo. Os 3 primeiros dizem quais são as permissões do usuário dono do arquivo. Os próximos 3 dizem quais são as permissões que os usuários do grupo possuem sobre o arquivo. Por fim, os 3 últimos caracteres dizem quais são as permissões para qualquer outra pessoa. Cada grupo de 3 caracteres versa sobre as permissões de leitura (r, read), escrita (w, write) e execução (x, execute) do arquivo, nesta ordem. O arquivo ilustrado na Listagem 7.5 possui permissão de leitura e escrita para o usuário *bob* e os membros do grupo *users*, os demais usuários possuem permissão apenas de leitura. Cada uma das permissões é binária (tem permissão ou não tem), desta forma cada conjunto de permissão expresso pelos 3 caracteres pode ser representado por um número binário com 3 bits. Desta forma, temos o seguinte:

<code>---</code>	$(000)_2 = 0$	todas permissões negadas
<code>--x</code>	$(001)_2 = 1$	permissão apenas de execução
<code>-w-</code>	$(010)_2 = 2$	permissão apenas de escrita
<code>-wx</code>	$(011)_2 = 3$	permissão para escrita e execução
<code>r--</code>	$(100)_2 = 4$	permissão para leitura
<code>r-x</code>	$(101)_2 = 5$	permissão para leitura e execução
<code>rw-</code>	$(110)_2 = 6$	permissão para leitura e escrita
<code>rwx</code>	$(111)_2 = 7$	permissão para leitura, escrita e execução

Podemos trocar o dono de um arquivo, assim como as permissões. Para tanto utilizaremos os comandos *chown* e *chmod*, respectivamente.

#### Listagem 7.6: Trocando o Dono de um Arquivo

```
$ chown john:users2 myfile  
$ ls -l myfile  
-rw-rw-r-- 1 john users2 375600 Dez 5 14:30 myfile
```

#### Listagem 7.7: Trocando as Permissões de um Arquivo

```
$ chmod 777 myfile  
$ ls -l myfile  
-rwxrwxrwx 1 john users2 375600 Dez 5 14:30 myfile  
$ chmod 600 myfile  
$ ls -l myfile  
-rw----- 1 john users2 375600 Dez 5 14:30 myfile
```

Podemos também mudar as permissões utilizando as representações simbólicas:

**u** para o usuário  
**g** para o grupo  
**o** para outros  
**a** para todos

**r** permissão de leitura  
**w** permissão de escrita  
**x** permissão de execução

Os seguintes exemplos na Listagem 7.8 ilustram algumas possíveis utilizações destas representações para alterar as permissões de um arquivo ou diretório.

#### Listagem 7.8: Exemplos de Utilização do chmod

```
# adicionar a permissao de execucao apenas ao usuario
```

```
$ chmod u+x arquivo  
  
# adicionar multiplas permissões (por exemplo, leitura e execução)  
$ chmod u+rwx arquivo  
  
# adicionar permissões diferentes a usuário (permissão de leitura) e (permissão de grupo)  
# grupo  
$ chmod u+r,g+x arquivo  
  
# remover permissões (leitura e execução)  
$ chmod u-rwx  
  
# adicionar a permissão de execução a todos os usuários  
$ chmod a+x
```

---

## 7.4 Comando chroot

O comando **chroot** roda um comando de outro diretório e trata aquele diretório como root. Este processo se chama de mudança de root e o novo diretório é chamado de **chroot jail**.

# Gerenciando Usuários

*O fonógrafo não tem nenhum valor comercial.*

Thomas Edison

## Sumário

8.1	Listando Todos os Usuários do Sistema . . . . .	70
8.2	Listando Grupos . . . . .	70
8.3	Adicionando Usuários . . . . .	70
8.4	Definindo Senha para Novos Usuários . . . . .	71
8.5	Apagando uma Conta de Usuário . . . . .	71
8.6	Adicionando um Novo Grupo . . . . .	72
8.7	Deletando um Grupo . . . . .	72

## 8.1 Listando Todos os Usuários do Sistema

Para listar todos os usuários do sistema basta utilizar o comando **dscl** como na Listagem 8.1.

### Listagem 8.1: Listando Todos os Usuários do Linux

```
musashi:$ dscl . list /Users | grep -v '^_'
alessandrovivas
daemon
macports
nobody
root
```

---

## 8.2 Listando Grupos

Para listar os grupos disponíveis utilize o comando **dscacheutil** como na Listagem 8.2.

### Listagem 8.2: Listando Todos os Usuários do Linux

```
musashi:$ dscacheutil -q group
name: _amavisd
password: *
gid: 83

name: _appleevents
password: *
gid: 55

name: _appowner
password: *
gid: 87

name: _appserveradm
password: *
gid: 81
users: alessandrovivas

name: _appserverusr
password: *
gid: 79
users: alessandrovivas

name: _appstore
password: *
gid: 33
...
...
```

---

## 8.3 Adicionando Usuários

Para adicionar um usuário basta utilizar o comando **dscl**. A sintaxe para adicionar usuário está apresentada na Listagem 8.3. Esta operação é privilegiada e só deve ser utilizada como **root**.

### Listagem 8.3: Adicionando Usuários

```
musashi-2:$ sudo dscl . create /Users/kika
Password:

% listando os usuários após comando
musashi-2:$ dscl . list /Users | grep -v '^_'
alessandrovivas
daemon
kika
macports
nobody
root
```

Você pode ainda inserir informações sobre este usuário como na Listagem 8.4 [4].

### Listagem 8.4: Definindo Informações Adicionais

```
dscl . -create /Users/kika
dscl . -create /Users/luser UserShell /bin/bash
dscl . -create /Users/luser RealName
dscl . -create /Users/luser UniqueID
dscl . -create /Users/luser PrimaryGroupID 100
dscl . -create /Users/luser NFSHomeDirectory /Users/kika
```

## 8.4 Definindo Senha para Novos Usuários

Para adicionar ou mudar a senha de um usuário basta utilizar o comando **passwd**. A sintaxe para adicionar usuário está apresentada na Listagem 8.5.

### Listagem 8.5: Definindo Senha de Usuário

```
musashi-2:$ sudo passwd kika
Changing password for kika.
New password:
Retype new password:
```

## 8.5 Apagando uma Conta de Usuário

Para apagar uma conta de usuário basta utilizar o comando **dscl**. A sintaxe para adicionar usuário está apresentada na Listagem 8.6 [15].

### Listagem 8.6: Apagando Usuários

```
musashi-2:$ sudo dscl . delete /Users/kika
Password:
musashi-2:$ dscl . list /Users | grep -v '^_'
alessandrovivas
daemon
macports
nobody
root
```

## 8.6 Adicionando um Novo Grupo

Para adicionar um grupo basta utilizar o comando **dscl**. A sintaxe para inserir um novo grupo está listado na Listagem 8.7 [5].

### Listagem 8.7: Adicionando um Novo Grupo

```
sudo dscl . create /Groups/labrador
sudo dscl . append /Groups/labrador gid 4200

% para listar o grupo
musashi-2:$ dscl . read /Groups/labrador
AppleMetaNodeLocation: /Local/Default
GeneratedUID: D0159845-D6AF-4FE9-90E4-66E86548AFAB
RecordName: labrador
RecordType: dsRecTypeStandard:Groups
```

Para adicionar mais informações utilize os comandos da Listagem 8.8 [8].

### Listagem 8.8: Comandos para Manipulação de Grupos

```
dscl . create /Groups/ladmins
dscl . create /Groups/ladmins RealName
dscl . create /Groups/ladmins passwd
dscl . create /Groups/ladmins gid 400
dscl . create /Groups/ladmins GroupMembership localadmin
dscl . append /Groups/ladmins GroupMembership 2ndlocaladmin
```

## 8.7 Deletando um Grupo

Para apagar um grupo basta utilizar o comando **dscl**. A sintaxe para apagar um grupo está listado na Listagem 8.9.

### Listagem 8.9: Apagando um Grupo

```
musashi-2:$ sudo dscl . delete /Groups/labrador
Password:
musashi-2:$ dscl . read /Groups/labrador
<dscl_cmd> DS Error: -14136 (eDSRecordNotFound)
```

# Comandos para Redes de Computadores

*O rádio não tem futuro.*

Lord Kevin, 1897

## Sumário

9.1 Comando hostname . . . . .	74
9.2 Comando curl . . . . .	74
9.3 Comando e Tabela ARP . . . . .	74
9.4 Listando Todas as Interfaces de Rede . . . . .	75
9.5 Trabalhando com o Endereço IP . . . . .	75
9.6 Comando ping . . . . .	79
9.7 Informações sobre Domínios . . . . .	80
9.8 Traçando caminhos de um host a outro . . . . .	81
9.9 Comando netstat . . . . .	82
9.10 Comando nettop . . . . .	93
9.11 Network Mapper . . . . .	93
9.12 Comando route . . . . .	96
9.13 Comando telnet . . . . .	97
9.14 Acesso Remoto com ssh . . . . .	101
9.15 Copiando Arquivos com scp . . . . .	101
9.16 Copiando um Diretório de um Servidor Remoto . . . . .	101
9.17 Comando tcpdump . . . . .	102
9.18 Baixando Sites com wget . . . . .	105

## 9.1 Comando hostname

Quer descobrir o nome de sua máquina? Uma maneira simples é entrar no terminal e conseguirá visualizar. Como exemplo, no meu terminal aparece o nome da máquina e o domínio que ela está vinculado **andarilho.local**. Você pode também utilizar o comando **hostname** apresentado na Listagem 9.1

### Listagem 9.1: Comando hostname

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ hostname  
andarilho.local
```

## 9.2 Comando curl

O comando **curl** tem várias utilidades. A Listagem 9.2 apresenta o comando para baixar uma página no terminal.

### Listagem 9.2: Download de uma Página

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ curl www.vivas.eng.br  
  
<!DOCTYPE HTML>  
<html lang pt-br>  
<head>  
  <meta charset=UTF-8>  
  <title>Alessandro Vivas Andrade</title>  
  <!-- Aqui chamamos o nosso arquivo css externo -->  
  <link rel=stylesheet type=text/css href=estilovivas.css />  
  <!--[if lte IE 8]>  
  <script src=http://html5shim.googlecode.com/svn/trunk/html5.js></script>  
  <![endif]-->  
</head>  
<body>  
  
....
```

Além desta funcionalidade ele pode baixar arquivos, fazer download de sites de ftp, postar no twitter, dentre outras. Uma boa referência é o site The Geek Stuff [16].

## 9.3 Comando e Tabela ARP

Em uma rede local precisamos do endereço MAC antes de qualquer comunicação e geralmente temos o endereço IP. Assim, cada computador possui um servidor ARP (Address Resolution Protocol) que fornece o endereço MAC de nossa máquina quando solicitado.

O protocolo ARP é responsável por receber um pacote com o endereço IP e enviar para o destinatário o endereço MAC. Quer saber a tabela ARP do seu computador? Isto é, os computadores que de alguma forma você entrou em contato? Este comando funciona no MAC, no Linux e até no Windows, Listagem 9.3. Através do comando **arp** é possível visualizar a tabela ARP.

### Listagem 9.3: Tabela ARP

```
musashi-2: alessandrovivas$ arp -a  
? (192.168.0.1) at 1c:7e:e5:46:92:e7 on en1 ifscope [ethernet]
```

```
? (192.168.0.107) at 70:56:81:97:14:31 on en1 ifscope permanent [ethernet]
? (192.168.0.109) at 70:62:b8:bf:fd:50 on en1 ifscope [ethernet]
? (192.168.0.150) at 70:62:b8:bf:fd:50 on en1 ifscope [ethernet]
```

---

## 9.4 Listando Todas as Interfaces de Rede

Para listar todas as interfaces de rede do utilize o comando **newtworksetup** da Listagem 9.4 [14].

### Listagem 9.4: Lista todas as Interfaces

```
musashi-2: alessandrovivas$ networksetup -listallhardwarereports

Hardware Port: Ethernet
Device: en0
Ethernet Address: 40:6c:8f:23:8d:84

Hardware Port: FireWire
Device: fw0
Ethernet Address: 40:6c:8f:ff:fe:7e:4d:0a

Hardware Port: Wi-Fi
Device: en1
Ethernet Address: 70:56:81:97:14:31

Hardware Port: Bluetooth PAN
Device: en3
Ethernet Address: 70:56:81:97:14:32

Hardware Port: Thunderbolt 1
Device: en2
Ethernet Address: d2:00:17:e4:d0:a0

Hardware Port: Thunderbolt Bridge
Device: bridge0
Ethernet Address: 42:6c:8f:32:5c:00

VLAN Configurations
=====
```

---

## 9.5 Trabalhando com o Endereço IP

Para verificar seu endereço IP basta utilizar o comando **ifconfig**. Como o computador está conectado na rede sem fio devemos verificar o endereço da interface **en1**. A Listagem 9.5 ilustra o resultado do comando **ifconfig**.

### 9.5.1 Verificando Endereço IP

#### Listagem 9.5: Verificando o Endereço IP

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
      options=3<RXCSUM,TXCSUM>
```

```

inet6 ::1 prefixlen 128
inet 127.0.0.1 netmask 0xffffffff
inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
    nd6 options=1<PERFORMNUD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=10b<RXCSUM,TXCSUM,VLAN_HWTAGGING,AV>
    ether 40:6c:8f:23:8d:84
    nd6 options=1<PERFORMNUD>
    media: autoselect (none)
    status: inactive
en1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether 70:56:81:97:14:31
    inet6 fe80::7256:81ff:fe97:1431%en1 prefixlen 64 scopeid 0x5
    inet 192.168.0.107 netmask 0xffffffff broadcast 192.168.0.255
    nd6 options=1<PERFORMNUD>
    media: autoselect
    status: active
en2: flags=963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX> mtu 1500
    options=60<TSO4,TSO6>
    ether d2:00:17:e4:d0:a0
    media: autoselect <full-duplex>
    status: inactive
fw0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 4078
    lladdr 40:6c:8f:ff:fe:7e:4d:0a
    nd6 options=1<PERFORMNUD>
    media: autoselect <full-duplex>
    status: inactive
p2p0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 2304
    ether 02:56:81:97:14:31
    media: autoselect
    status: inactive
bridge0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=63<RXCSUM,TXCSUM,TSO4,TSO6>
    ether 42:6c:8f:32:5c:00
    Configuration:
        id 0:0:0:0:0:0 priority 0 hellotime 0 fwddelay 0
        maxage 0 holdcnt 0 proto stp maxaddr 100 timeout 1200
        root id 0:0:0:0:0:0 priority 0 ifcost 0 port 0
        ipfilter disabled flags 0x2
    member: en2 flags=3<LEARNING,DISCOVER>
        ifmaxaddr 0 port 6 priority 0 path cost 0
    nd6 options=1<PERFORMNUD>
    media: <unknown type>
    status: inactive
utun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1380
    inet6 fe80::415d:c924:8387:28c1%utun0 prefixlen 64 scopeid 0xa
    inet6 fdd6:d35e:5451:e50:415d:c924:8387:28c1 prefixlen 64
    nd6 options=1<PERFORMNUD>

```

Para verificar o endereço IP de uma interface específica utilize o comando 9.6.

**Listagem 9.6: Verificando o Endereço de uma Interface Específica**

```

musashi-2: alessandrovivas$ ifconfig en1
en1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether 70:56:81:97:14:31
    inet6 fe80::7256:81ff:fe97:1431%en1 prefixlen 64 scopeid 0x5
    inet 192.168.0.107 netmask 0xffffffff broadcast 192.168.0.255
    nd6 options=1<PERFORMNUD>
    media: autoselect

```

```
status: active
```

---

### 9.5.2 Habilitando e Desabilitando a Interface de Rede

Para habilitar ou desabilitar uma interface de rede utilizamos o comando **ifconfig**. Primeiro vamos verificar o status da interface de rede sem fio, Listagem 9.7.

#### Listagem 9.7: Verificando Status da Rede Ethernet

```
musashi-2: alessandrovivas$ ifconfig en1
en1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      ether 70:56:81:97:14:31
      inet6 fe80::7256:81ff:fe97:1431%en1 prefixlen 64 scopeid 0x5
          inet 192.168.0.107 netmask 0xffffffff broadcast 192.168.0.255
              nd6 options=1<PERFORMNUD>
              media: autoselect
              status: active
```

---

Para desabilitar utilizamos o comando da Listagem 9.7. Para verificar vamos utilizar o comando **ifconfig** na Listagem 9.8.

#### Listagem 9.8: Desabilitando a Interface de Rede Ethernet

```
musashi-2: alessandrovivas$ sudo ifconfig en1 down
Password:
```

---

Para verificar se a interface está desabilitada, proceda com o comando **ifconfig** como na Listagem 9.9.

#### Listagem 9.9: Verificando a Ação Realizada na Listagem 9.8

```
musashi-2: alessandrovivas$ ifconfig en1
en1: flags=8822<BROADCAST,SMART,SIMPLEX,MULTICAST> mtu 1500
      ether 70:56:81:97:14:31
      nd6 options=1<PERFORMNUD>
      media: autoselect (<unknown type>)
      status: inactive
```

---

Para habilitar, utilizamos o comando da Listagem 9.10 e, para verificar, vamos utilizar o comando **ifconfig** na Listagem 9.11.

#### Listagem 9.10: Habilitando a Interface de Rede Ethernet

```
musashi-2: alessandrovivas$ sudo ifconfig en1 up
```

---

#### Listagem 9.11: Verificando a Ação Realizada na Listagem 9.10

```
musashi-2: alessandrovivas$ ifconfig en1
en1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      ether 70:56:81:97:14:31
      inet6 fe80::7256:81ff:fe97:1431%en1 prefixlen 64 scopeid 0x5
          inet 192.168.0.107 netmask 0xffffffff broadcast 192.168.0.255
              nd6 options=1<PERFORMNUD>
              media: autoselect
              status: active
```

---

### 9.5.3 Alterando a MTU de uma Interface

MTU (*Maximum Transmission Unit*) é o tamanho do maior datagrama que pode ser transmitido em uma determinada rede. As redes Ethernet utilizam o tamanho de 1500 bytes. Para listar a MTU utilizada usamos o comando **ifconfig** da Listagem 9.12. Para verificar isto utilizamos o comando **ifconfig**.

#### Listagem 9.12: Verificando MTU da Rede Ethernet

```
musashi-2: alessandrovivas$ ifconfig en1
en1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
ether 70:56:81:97:14:31
inet6 fe80::7256:81ff:fe97:1431%en1 prefixlen 64 scopeid 0x5
inet 192.168.0.107 netmask 0xffffffff00 broadcast 192.168.0.255
nd6 options=1<PERFORMNUD>
media: autoselect
status: active
```

---

É possível alterar a MTU utilizando o comando **ifconfig** da Listagem 9.13.

#### Listagem 9.13: Alterando o MTU da Placa de Rede

```
musashi-2: alessandrovivas$ sudo ifconfig en1 mtu 1000
musashi-2: alessandrovivas$ ifconfig en1
en1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1000
ether 70:56:81:97:14:31
inet6 fe80::7256:81ff:fe97:1431%en1 prefixlen 64 scopeid 0x5
inet 192.168.0.107 netmask 0xffffffff00 broadcast 192.168.0.255
nd6 options=1<PERFORMNUD>
media: autoselect
status: active
```

---

### 9.5.4 Alterando Endereço IP

Se quiser alterar seu endereço IP basta utilizar o comando **ifconfig**. O comando recebe os seguintes parâmetros: interface, endereço IP e máscara. A Listagem 9.14 apresenta o endereço IP atual da interface eth0.

#### Listagem 9.14: Verificando o Endereço IP

```
musashi-2: alessandrovivas$ ifconfig en1
en1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
ether 70:56:81:97:14:31
inet6 fe80::7256:81ff:fe97:1431%en1 prefixlen 64 scopeid 0x5
inet 192.168.0.107 netmask 0xffffffff00 broadcast 192.168.0.255
nd6 options=1<PERFORMNUD>
media: autoselect
status: active
```

---

A Listagem 9.15 apresenta o comando de alteração do endereço IP.

#### Listagem 9.15: Alterando Endereço IP

```
musashi-2: alessandrovivas$ sudo ifconfig en1 192.168.0.108 netmask 255.255.255.0
musashi-2: alessandrovivas$ ifconfig en1
en1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
ether 70:56:81:97:14:31
inet6 fe80::7256:81ff:fe97:1431%en1 prefixlen 64 scopeid 0x5
inet 192.168.0.108 netmask 0xffffffff00 broadcast 192.168.0.255
nd6 options=1<PERFORMNUD>
media: autoselect
status: active
```

### 9.5.5 Descobrir endereço IP de um Determinado Host

Muitas vezes precisamos descobrir o endereço IP de um determinado host. Para realizar esta tarefa utilizamos o comando **host**. A Listagem 9.16 mostra como descobrir o endereço IP do Google.

#### Listagem 9.16: Descobrindo o Endereço Ip de um Host

```
musashi-2: alessandrovivas$ host www.google.com
www.google.com has address 173.194.118.51
www.google.com has address 173.194.118.48
www.google.com has address 173.194.118.52
www.google.com has address 173.194.118.49
www.google.com has address 173.194.118.50
www.google.com has IPv6 address 2800:3f0:4001:811::1012
```

## 9.6 Comando ping

O comando **ping** serve para fazer verificações sobre o status de funcionamento de computadores em uma rede. Com ele podemos medir o tempo de ida e volta (*round time trip*) que um pacote demora para ir do seu host para outro. Você pode usar tanto o endereço IP do host ou o endereço Web. A Listagem 9.17 mostra o funcionamento do comando **ping**. Podemos passar um endereço IP, como ilustrado na Listagem 9.17, ou utilizar como argumento um endereço Web, como ilustrado na Listagem 9.18. Para interromper o comando basta digita <Ctrl+C>. Quando interromper o comando serão mostradas as estatísticas dos testes realizados.

#### Listagem 9.17: Comando ping

```
musashi-2: alessandrovivas$ ping 192.168.0.1
PING 192.168.0.1 (192.168.0.1): 56 data bytes
64 bytes from 192.168.0.1: icmp_seq=0 ttl=64 time=2.423 ms
64 bytes from 192.168.0.1: icmp_seq=1 ttl=64 time=4.154 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=64 time=2.690 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=64 time=1.941 ms
64 bytes from 192.168.0.1: icmp_seq=4 ttl=64 time=1.912 ms
64 bytes from 192.168.0.1: icmp_seq=5 ttl=64 time=1.976 ms
64 bytes from 192.168.0.1: icmp_seq=6 ttl=64 time=1.922 ms
64 bytes from 192.168.0.1: icmp_seq=7 ttl=64 time=4.742 ms
64 bytes from 192.168.0.1: icmp_seq=8 ttl=64 time=5.552 ms
^C
--- 192.168.0.1 ping statistics ---
9 packets transmitted, 9 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.912/3.035/5.552/1.326 ms
```

#### Listagem 9.18: Exemplo do Comando ping

```
musashi-2: alessandrovivas$ ping www.ufsj.edu.br
PING www.ufsj.edu.br (200.17.67.239): 56 data bytes
64 bytes from 200.17.67.239: icmp_seq=0 ttl=45 time=43.712 ms
64 bytes from 200.17.67.239: icmp_seq=1 ttl=45 time=42.738 ms
64 bytes from 200.17.67.239: icmp_seq=2 ttl=45 time=51.829 ms
64 bytes from 200.17.67.239: icmp_seq=3 ttl=45 time=45.914 ms
64 bytes from 200.17.67.239: icmp_seq=4 ttl=45 time=44.285 ms
64 bytes from 200.17.67.239: icmp_seq=5 ttl=45 time=49.944 ms
^C
--- www.ufsj.edu.br ping statistics ---
6 packets transmitted, 6 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 42.738/46.404/51.829/3.351 ms
```

---

Você pode utilizar a opção **-c** com o **ping** onde especifica o número de pacotes enviados pelo ping. A Listagem 9.19 apresenta o comando para 10 tentativas.

#### Listagem 9.19: Comando ping

```
musashi-2: alessandrovivas$ ping -c 10 www.ufsj.edu.br
PING www.ufsj.edu.br (200.17.67.239): 56 data bytes
64 bytes from 200.17.67.239: icmp_seq=0 ttl=44 time=171.934 ms
64 bytes from 200.17.67.239: icmp_seq=1 ttl=44 time=173.687 ms
64 bytes from 200.17.67.239: icmp_seq=2 ttl=44 time=182.566 ms
64 bytes from 200.17.67.239: icmp_seq=3 ttl=44 time=164.998 ms
64 bytes from 200.17.67.239: icmp_seq=4 ttl=44 time=199.799 ms
64 bytes from 200.17.67.239: icmp_seq=5 ttl=44 time=167.757 ms
64 bytes from 200.17.67.239: icmp_seq=6 ttl=44 time=172.373 ms
64 bytes from 200.17.67.239: icmp_seq=7 ttl=44 time=170.344 ms
64 bytes from 200.17.67.239: icmp_seq=8 ttl=44 time=189.462 ms
64 bytes from 200.17.67.239: icmp_seq=9 ttl=44 time=188.202 ms

--- www.ufsj.edu.br ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 164.998/178.112/199.799/10.729 ms
```

---

Quer controlar o intervalo de tempo entre os pacotes enviados? Utilize a opção **-i** como mostrado na Listagem 9.20, vamos aproveitar e utilizar a opção de enviar 3 pacotes.

#### Listagem 9.20: Comando ping com Opção de Tempo

```
musashi-2: alessandrovivas$ ping -i 5 -c 3 192.168.0.1
PING 192.168.0.1 (192.168.0.1): 56 data bytes
64 bytes from 192.168.0.1: icmp_seq=0 ttl=64 time=1.948 ms
64 bytes from 192.168.0.1: icmp_seq=1 ttl=64 time=2.609 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=64 time=1.982 ms

--- 192.168.0.1 ping statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.948/2.180/2.609/0.304 ms
```

## 9.7 Informações sobre Domínios

### 9.7.1 Comando dig

Uma maneira de obter informações sobre domínios é utilizar o comando **dig**. A sintaxe é bem simples, pois basta usar como parâmetro o domínio desejado. A Listagem 9.21 ilustra a utilização do comando.

#### Listagem 9.21: Descobrir Informações sobre um Domínio

```
musashi-2: alessandrovivas$ dig www.vivas.eng.br

; <>> DiG 9.8.3-P1 <>> www.vivas.eng.br
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 5622
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
```

```

;www.vivas.eng.br.      IN  A

;; ANSWER SECTION:
www.vivas.eng.br. 14098 IN  CNAME vivas.eng.br.
vivas.eng.br.      14098 IN  A 158.69.55.227

;; Query time: 27 msec
;; SERVER: 192.168.0.1#53(192.168.0.1)
;; WHEN: Thu Nov 26 16:16:22 2015
;; MSG SIZE  rcvd: 76

```

---

### 9.7.2 Comando nslookup

Outra maneira é utilizar o comando **nslookup**. A Listagem 9.22 ilustra a utilização do comando.

#### Listagem 9.22: Utilizando o comando nslookup

```

musashi-2: alessandrovivas$ nslookup www.vivas.eng.br
Server:  192.168.0.1
Address: 192.168.0.1#53

Non-authoritative answer:
www.vivas.eng.br canonical name = vivas.eng.br.
Name:  vivas.eng.br
Address: 158.69.55.227

```

---

## 9.8 Traçando caminhos de um host a outro

O comando **traceroute** é uma ferramenta para imprimir os caminhos de seu host até um destino. Ele mostra todos os roteadores que o pacote enviado passa e imprime informações sobre o tempo decorrido.

Para usar basta inserir o destino como na Listagem 9.23. Alguns roteadores são programados para não responder e por isto aparecem o símbolo \*. Outra situação é ocorrência de perda de pacotes.

#### Listagem 9.23: Rotas

```

musashi-2: alessandrovivas$ traceroute www.ufmg.br
traceroute to www.ufmg.br (150.164.250.1), 64 hops max, 52 byte packets
 1  192.168.0.1 (192.168.0.1)  2.715 ms  4.439 ms  3.417 ms
 2  192.168.2.1 (192.168.2.1)  3.346 ms  3.704 ms  3.498 ms
 3  10.1.223.1 (10.1.223.1)  7.234 ms  7.911 ms  7.566 ms
 4  10.12.0.9 (10.12.0.9)  19.476 ms  8.853 ms  6.252 ms
 5  10.12.0.5 (10.12.0.5)  10.463 ms  8.325 ms  13.822 ms
 6  10.12.0.53 (10.12.0.53)  13.158 ms  18.022 ms  10.950 ms
 7  177.124.74.101 (177.124.74.101)  15.709 ms  20.677 ms  15.637 ms
 8  205-108-107-179.telbrax.net.br (179.107.108.205)  30.665 ms  25.503 ms  27.071 ms
 9  xe-0-0-0-rtcr01-ste-bhe.telbrax.net.br (187.1.92.5)  37.461 ms  35.347 ms  31.618 ms
10  xe-0-0-3-rtcrag01-ctr-rjo.telbrax.net.br (187.1.92.18)  69.738 ms  41.952 ms  40.461 ms
11  as1916.rj.ix.br (200.219.138.101)  34.731 ms  40.536 ms  28.239 ms
12  ce-rj.bkb.rnp.br (200.143.253.25)  136.318 ms  131.534 ms  137.465 ms
13  200.143.253.150 (200.143.253.150)  122.523 ms  128.715 ms  133.530 ms
14  mg-ce-oi.bkb.rnp.br (200.143.252.141)  162.010 ms  164.365 ms  168.518 ms
15  mg-mg2.bkb.rnp.br (200.143.253.225)  178.869 ms  169.892 ms  167.259 ms
16  lanmg-mg.bkb.rnp.br (200.143.253.162)  166.548 ms  163.601 ms  163.606 ms
17  bombom.pop-mg.rnp.br (200.131.0.14)  163.064 ms  166.751 ms  168.540 ms

```

```
18 pop-mg-novo.rede.ufmg.br (200.131.2.220) 179.963 ms 174.378 ms 164.688 ms
19 pop.central-core.rede.ufmg.br (150.164.164.149) 167.007 ms 164.138 ms 169.452 ms
20 cecom-gw.rede.ufmg.br (150.164.255.31) 172.071 ms 172.544 ms 170.190 ms
21 www.ufmg.br (150.164.250.1) 170.792 ms !Z 165.759 ms !Z 178.516 ms !Z
```

---

### 9.8.1 Descobrindo o Endereço do seu Roteador sem Fio

Uma das maneiras é utilizar o comando traceroute. O primeiro salto sempre será o IP do seu roteador sem fio. Verifique a Listagem 9.24. Neste caso enviamos o comando para dois destinos diferentes e observamos que os dois sempre no primeiro salto passam pelo o IP 192.168.0.1 que é o do roteador sem fio.

#### Listagem 9.24: Endereço do Roteador sem Fio

```
musashi-2: alessandrovivas$ traceroute www.ufmg.br
traceroute to www.ufmg.br (150.164.250.1), 64 hops max, 52 byte packets
 1  192.168.0.1 (192.168.0.1)  2.715 ms  4.439 ms  3.417 ms
 2  192.168.2.1 (192.168.2.1)  3.346 ms  3.704 ms  3.498 ms
 3  10.1.223.1 (10.1.223.1)  7.234 ms  7.911 ms  7.566 ms
 4  10.12.0.9 (10.12.0.9)  19.476 ms  8.853 ms  6.252 ms
 5  10.12.0.5 (10.12.0.5)  10.463 ms  8.325 ms  13.822 ms
 6  10.12.0.53 (10.12.0.53)  13.158 ms  18.022 ms  10.950 ms
 7  177.124.74.101 (177.124.74.101)  15.709 ms  20.677 ms  15.637 ms
 8  205-108-107-179.telbrax.net.br (179.107.108.205)  30.665 ms  25.503 ms  27.071 ms
 9  xe-0-0-0-rtcr01-ste-bhe.telbrax.net.br (187.1.92.5)  37.461 ms  35.347 ms  31.618 ms
10  xe-0-0-3-rtcrag01-ctr-rjo.telbrax.net.br (187.1.92.18)  69.738 ms  41.952 ms  40.461
    ms
11  as1916.rj.ix.br (200.219.138.101)  34.731 ms  40.536 ms  28.239 ms
12  ce-rj.bkb.rnp.br (200.143.253.25)  136.318 ms  131.534 ms  137.465 ms
13  200.143.253.150 (200.143.253.150)  122.523 ms  128.715 ms  133.530 ms
14  mg-ce-oi.bkb.rnp.br (200.143.252.141)  162.010 ms  164.365 ms  168.518 ms
15  mg-mg2.bkb.rnp.br (200.143.253.225)  178.869 ms  169.892 ms  167.259 ms
16  lanmg-mg.bkb.rnp.br (200.143.253.162)  166.548 ms  163.601 ms  163.606 ms
17  bombom.pop-mg.rnp.br (200.131.0.14)  163.064 ms  166.751 ms  168.540 ms
18  pop-mg-novo.rede.ufmg.br (200.131.2.220)  179.963 ms  174.378 ms  164.688 ms
19  pop.central-core.rede.ufmg.br (150.164.164.149)  167.007 ms  164.138 ms  169.452 ms
20  cecom-gw.rede.ufmg.br (150.164.255.31)  172.071 ms  172.544 ms  170.190 ms
21  www.ufmg.br (150.164.250.1)  170.792 ms !Z  165.759 ms !Z  178.516 ms !Z
```

---

## 9.9 Comando netstat

O comando **netstat** é uma ferramenta essencial para administradores de rede. Ele possibilita fazer rastreamento das portas que são utilizadas no seu computador. A Listagem 9.25 apresenta estatística dos protocolos.

#### Listagem 9.25: Estatísticas de Rede com netstat

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ netstat -s
tcp:
 1925398 packets sent
 1607356 data packets (1829541213 bytes)
 32008 data packets (34324810 bytes) retransmitted
 1 resend initiated by MTU discovery
 226064 ack-only packets (1121 delayed)
 0 URG only packet
 0 window probe packet
 35540 window update packets
```

```
24993 control packets
0 data packet sent after flow control
1935824 checksummed in software
    1935824 segments (1926500613 bytes) over IPv4
        0 segment (0 byte) over IPv6
1363817 packets received
    920337 acks (for 1829070799 bytes)
    214225 duplicate acks
    0 ack for unsent data
    185279 packets (181078163 bytes) received in-sequence
    1691 completely duplicate packets (886235 bytes)
    73 old duplicate packets
    0 received packet dropped due to low memory
    21 packets with some dup. data (3415 bytes duped)
    36793 out-of-order packets (49127124 bytes)
    0 packet (0 byte) of data after window
    0 window probe
    7197 window update packets
    800 packets received after close
    26 bad resets
    0 discarded for bad checksum
    1362946 checksummed in software
        1362946 segments (304731494 bytes) over IPv4
            0 segment (0 byte) over IPv6
        0 discarded for bad header offset field
        0 discarded because packet too short
16986 connection requests
14 connection accepts
0 bad connection attempt
0 listen queue overflow
8112 connections established (including accepts)
17966 connections closed (including 1578 drops)
    206 connections updated cached RTT on close
    206 connections updated cached RTT variance on close
    139 connections updated cached ssthresh on close
1987 embryonic connections dropped
914978 segments updated rtt (of 233061 attempts)
65076 retransmit timeouts
    737 connections dropped by rexmit timeout
    0 connection dropped after retransmitting FIN
0 persist timeout
    0 connection dropped by persist timeout
10121 keepalive timeouts
    8990 keepalive probes sent
    2 connections dropped by keepalive
172843 correct ACK header predictions
166188 correct data packet header predictions
12589 SACK recovery episodes
19870 segment retransmits in SACK recovery episodes
24828175 byte retransmits in SACK recovery episodes
222866 SACK options (SACK blocks) received
36474 SACK options (SACK blocks) sent
0 SACK scoreboard overflow
0 LRO coalesced packet
    0 time LRO flow table was full
    0 collision in LRO flow table
    0 time LRO coalesced 2 packets
    0 time LRO coalesced 3 or 4 packets
    0 time LRO coalesced 5 or more packets
15035 limited transmits done
4419 early retransmits done
```

```
292 times cumulative ack advanced along with SACK
2316 probe timeouts
    783 times retransmit timeout triggered after probe
    0 time probe packets were sent for an interface
    0 time couldn't send probe packets for an interface
    79 times fast recovery after tail loss
    849 times recovered last packet
    2640 SACK based rescue retransmits
0 client connection attempted to negotiate ECN
0 client connection successfully negotiated ECN
0 time graceful fallback to Non-ECN connection
0 time lost ECN negotiating SYN, followed by retransmission
0 server connection attempted to negotiate ECN
0 server connection successfully negotiated ECN
0 time lost ECN negotiating SYN-ACK, followed by retransmission
0 time received congestion experienced (CE) notification
0 time CWR was sent in response to ECE
0 time sent ECE notification
0 connection received CE atleast once
0 connection received ECE atleast once
0 connection using ECN have seen packet loss but no CE
0 connection using ECN have seen packet loss and CE
0 connection using ECN received CE but no packet loss
187 times packet reordering was detected on a connection
    6649 times transmitted packets were reordered
    1530 times fast recovery was delayed to handle reordering
    6649 times retransmission was avoided by delaying recovery
    0 retransmission not needed
1263 times DSACK option was sent
    9071 times DSACK option was received
    124 times DSACK was disabled on a connection
    1920 times recovered from bad retransmission using DSACK
    101 times ignored DSACK due to ack loss
    0 time ignored old DSACK options
37 times PMTU Blackhole detection, size reverted
0 connection were dropped after long sleep
0 time a TFO-cookie has been announced
0 SYN with data and a valid TFO-cookie have been received
0 SYN with TFO-cookie-request received
0 time an invalid TFO-cookie has been received
2 times we requested a TFO-cookie
    0 time the peer announced a TFO-cookie
0 time we combined SYN with data and a TFO-cookie
    0 time our SYN with data has been acknowledged
0 time a connection-attempt with TFO fell back to regular TCP
    0 time a TFO-connection blackhole'd
udp:
214219 datagrams received
    0 with incomplete header
    0 with bad data length field
    0 with bad checksum
    29 with no checksum
211987 checksummed in software
    211667 datagrams (60180966 bytes) over IPv4
    320 datagrams (78621 bytes) over IPv6
275 dropped due to no socket
72 broadcast/multicast datagrams undelivered
0 time multicast source filter matched
84 dropped due to full socket buffers
0 not for hashed pcb
213788 delivered
```

```
210793 datagrams output
 210407 checksummed in software
 210046 datagrams (118677799 bytes) over IPv4
 361 datagrams (54732 bytes) over IPv6
ip:
 1585896 total packets received
 0 bad header checksum
 1582888 headers (31661484 bytes) checksummed in software
 0 with size smaller than minimum
 0 with data size < data length
 0 with data size > data length
 0 packet forced to software checksum
 0 with ip length > max ip packet size
 0 with header length < data size
 0 with data length < header length
 0 with bad options
 0 with incorrect version number
 0 fragment received
 0 dropped (dup or out of space)
 0 dropped after timeout
 0 reassembled ok
 1584750 packets for this host
 951 packets for unknown/unsupported protocol
 0 packet forwarded (0 packet fast forwarded)
 40 packets not forwardable
 155 packets received for unknown multicast group
 0 redirect sent
 1208 input packets not chained due to collision
 99327 input packets processed in a chain
 315 input packets unable to chain
 4723 input packet chains processed with length greater than 2
 681 input packet chains processed with length greater than 4
 1485046 input packets did not go through list processing path
 2149482 packets sent from this host
 130 packets sent with fabricated ip header
 0 output packet dropped due to no bufs, etc.
 12 output packets discarded due to no route
 1 output datagram fragmented
 2 fragments created
 1 datagram that can't be fragmented
 0 tunneling packet that can't find gif
 67 datagrams with bad address in header
 0 packet dropped due to no bufs for control data
 2148214 headers (42969812 bytes) checksummed in software
icmp:
 275 calls to icmp_error
 0 error not generated 'cuz old message was icmp
Output histogram:
  destination unreachable: 275
 0 message with bad code fields
 0 message < minimum length
 0 bad checksum
 0 message with bad length
 0 multicast echo requests ignored
 0 multicast timestamp requests ignored
Input histogram:
  echo reply: 28
  destination unreachable: 7017
  time exceeded: 208
 0 message response generated
  ICMP address mask responses are disabled
```

```
igmp:
    922 messages received
    0 message received with too few bytes
    0 message received with wrong TTL
    0 message received with bad checksum
    538 V1/V2 membership queries received
    0 V3 membership queries received
    0 membership queries received with invalid field(s)
    538 general queries received
    0 group queries received
    0 group-source queries received
    0 group-source queries dropped
    384 membership reports received
    0 membership report received with invalid field(s)
    384 membership reports received for groups to which we belong
    0 V3 report received without Router Alert
    1343 membership reports sent

ipsec:
    0 inbound packet processed successfully
    0 inbound packet violated process security policy
    0 inbound packet with no SA available
    0 invalid inbound packet
    0 inbound packet failed due to insufficient memory
    0 inbound packet failed getting SPI
    0 inbound packet failed on AH replay check
    0 inbound packet failed on ESP replay check
    0 inbound packet considered authentic
    0 inbound packet failed on authentication
    0 outbound packet processed successfully
    0 outbound packet violated process security policy
    0 outbound packet with no SA available
    0 invalid outbound packet
    0 outbound packet failed due to insufficient memory
    0 outbound packet with no route

arp:
    183 ARP requests sent
    1750 ARP replies sent
    0 ARP announcement sent
    51659 ARP requests received
    43 ARP replies received
    51702 total ARP packets received
    0 ARP conflict probe sent
    0 invalid ARP resolve request
    0 total packet dropped due to lack of memory
    193 total packets dropped due to no ARP entry
    8 total packets dropped during ARP entry removal
    138 ARP entries timed out
    1 Duplicate IP seen

mptcp:
    0 data packet sent
    0 data byte sent
    0 data packet received
    0 data byte received
    0 packet with an invalid MPCAP option
    0 packet with an invalid MPJOIN option
    0 time primary subflow fell back to TCP
    0 time secondary subflow fell back to TCP
    0 DSS option drop
    0 other invalid MPTCP option
    0 time the MPTCP subflow window was reduced
    0 bad DSS checksum
```

```
0 time received out of order data
0 subflow switch
0 subflow switche due to advisory
0 subflow switche due to rtt
0 subflow switche due to rto
0 subflow switche due to peer
0 number of subflow probe
ip6:
5185 total packets received
 0 with size smaller than minimum
 0 with data size < data length
 0 with data size > data length
    0 packet forced to software checksum
 0 with bad options
 0 with incorrect version number
 0 fragment received
    0 dropped (dup or out of space)
    0 dropped after timeout
    0 exceeded limit
    0 reassembled ok
 612 packets for this host
 0 packet forwarded
 4573 packets not forwardable
 0 redirect sent
 4573 multicast packets which we don't join
 0 packet whose headers are not continuous
 0 tunneling packet that can't find gif
 0 packet discarded due to too many headers
 0 forward cache hit
 0 forward cache miss
 0 packet dropped due to no bufs for control data
217 packets sent from this host
 0 packet sent with fabricated ip header
 0 output packet dropped due to no bufs, etc.
 6152 output packets discarded due to no route
 0 output datagram fragmented
 0 fragment created
 0 datagram that can't be fragmented
 0 packet that violated scope rules
Input histogram:
 hop by hop: 66
 UDP: 4658
 ICMP6: 461
Mbuf statistics:
 0 one mbuf
 two or more mbuf:
  lo0= 190
 4995 one ext mbuf
 0 two or more ext mbuf
 0 failure of source address selection
source addresses on an outgoing I/F
 0 addresses scope=0
 0 node-local
 0 link-local
 0 addresses scope=3
 0 addresses scope=4
 0 site-local
 0 addresses scope=6
 0 addresses scope=7
 0 addresses scope=8
 0 addresses scope=9
```

```
0 addresses scope=a
0 addresses scope=b
0 addresses scope=c
0 addresses scope=d
0 global
0 addresses scope=f
source addresses on a non-outgoing I/F
0 addresses scope=0
0 node-local
0 link-local
0 addresses scope=3
0 addresses scope=4
0 site-local
0 addresses scope=6
0 addresses scope=7
0 addresses scope=8
0 addresses scope=9
0 addresses scope=a
0 addresses scope=b
0 addresses scope=c
0 addresses scope=d
0 global
0 addresses scope=f
source addresses of same scope
0 addresses scope=0
0 node-local
0 link-local
0 addresses scope=3
0 addresses scope=4
0 site-local
0 addresses scope=6
0 addresses scope=7
0 addresses scope=8
0 addresses scope=9
0 addresses scope=a
0 addresses scope=b
0 addresses scope=c
0 addresses scope=d
0 global
0 addresses scope=f
source addresses of a different scope
0 addresses scope=0
0 node-local
0 link-local
0 addresses scope=3
0 addresses scope=4
0 site-local
0 addresses scope=6
0 addresses scope=7
0 addresses scope=8
0 addresses scope=9
0 addresses scope=a
0 addresses scope=b
0 addresses scope=c
0 addresses scope=d
0 global
0 addresses scope=f
deprecated source addresses
0 addresses scope=0
0 node-local
0 link-local
```

```
0 addresses scope=3
0 addresses scope=4
0 site-local
0 addresses scope=6
0 addresses scope=7
0 addresses scope=8
0 addresses scope=9
0 addresses scope=a
0 addresses scope=b
0 addresses scope=c
0 addresses scope=d
0 global
0 addresses scope=f
source addresse selection
197 rules default
0 rule prefer same address
0 rule prefer appropriate scope
0 rule avoid deprecated addresses
0 rule prefer home addresses
0 rule prefer outgoing interface
0 rule prefer addresses in a prefix advertised by the next-hop
20 rules prefer matching label
0 rule prefer temporary addresses
0 rule prefer addresses on alive interfaces
0 rule use longest matching prefix
0 duplicate address detection collision
0 times ignored source on secondary expensive I/F
icmp6:
0 call to icmp_error
0 error not generated because old message was icmp error or so
0 error not generated because rate limitation
Output histogram:
router solicitation: 9
neighbor solicitation: 5
neighbor advertisement: 1
MLDv2 listener report: 9
0 message with bad code fields
0 message < minimum length
22 bad checksums
0 message with bad length
Input histogram:
router advertisement: 78
neighbor solicitation: 1
neighbor advertisement: 1
Histogram of error messages to be generated:
0 no route
0 administratively prohibited
0 beyond scope
0 address unreachable
0 port unreachable
0 packet too big
0 time exceed transit
0 time exceed reassembly
0 erroneous header field
0 unrecognized next header
0 unrecognized option
0 redirect
0 unknown
0 message response generated
0 message with too many ND options
0 message with bad ND options
```

```
0 bad neighbor solicitation message
1 bad neighbor advertisement message
0 bad router solicitation message
0 bad router advertisement message
0 bad redirect message
0 path MTU change
ipsec6:
0 inbound packet processed successfully
0 inbound packet violated process security policy
0 inbound packet with no SA available
0 invalid inbound packet
0 inbound packet failed due to insufficient memory
0 inbound packet failed getting SPI
0 inbound packet failed on AH replay check
0 inbound packet failed on ESP replay check
0 inbound packet considered authentic
0 inbound packet failed on authentication
0 outbound packet processed successfully
0 outbound packet violated process security policy
0 outbound packet with no SA available
0 invalid outbound packet
0 outbound packet failed due to insufficient memory
0 outbound packet with no route
rip6:
0 message received
0 checksum calculation on inbound
0 message with bad checksum
0 message dropped due to no socket
0 multicast message dropped due to no socket
0 message dropped due to full socket buffers
0 delivered
0 datagram output
pfkey:
12 requests sent to userland
1232 bytes sent to userland
histogram by message type:
register: 3
x_spdadd: 4
x_spddelete: 4
x_spddump: 1
0 message with invalid length field
0 message with invalid version field
0 message with invalid message type field
0 message too short
0 message with memory allocation failure
0 message with duplicate extension
0 message with invalid extension type
0 message with invalid sa type
0 message with invalid address extension
15 requests sent from userland
2016 bytes sent from userland
histogram by message type:
register: 3
x_spdadd: 4
x_spddelete: 4
x_spddump: 4
0 message toward single socket
0 message toward all sockets
0 message toward registered sockets
0 message with memory allocation failure
kevt:
```

```

20 current kernel control sockets
1112 kernel control generation count
0 bad vendor failure
1271 message too big failures
0 out of memory failure
0 message dropped due to full socket buffers
12859 messages posted
kctl:
0 total kernel control module registered
12 current kernel control modules registered
41 current kernel control sockets
14271 kernel control generation count
7150 connection attempts
0 connection failure
27 send failures
0 send list failure
0 enqueue failure
0 packet dropped due to full socket buffers
xbkidle:
1 max per process
600 maximum time (seconds)
131072 high water mark
0 socket option not supported failure
0 too many sockets failure
0 total socket requested OK
0 extended bk idle socket
0 no cellular failure
0 no time failures
0 forced defunct socket
0 resumed socket
0 timeout expired failure
0 timer rescheduled
0 no delegated failure

```

---

A Listagem 9.26 apresenta a maneira de obter estatísticas das interfaces de rede.

#### Listagem 9.26: Comando netstat -i

Name	Mtu	Network	Address	Ipkts	Ierrrs	Opkts	Oerrs	Coll
lo0	16384	<Link#1>		3204	0	3204	0	0
lo0	16384	localhost	::1	3204	-	3204	-	-
lo0	16384	127	localhost	3204	-	3204	-	-
lo0	16384	localhost	fe80:1::1	3204	-	3204	-	-
gif0*	1280	<Link#2>		0	0	0	0	0
stf0*	1280	<Link#3>		0	0	0	0	0
eno0	1500	<Link#4>	40:6c:8f:23:8d:84	0	0	0	0	0
en1	1500	<Link#5>	70:56:81:97:14:31	1646278	0	2156147	0	0
en1	1500	musashi-2.1	fe80:5::7256:81ff	1646278	-	2156147	-	-
en1	1500	192.168.0	192.168.0.108	1646278	-	2156147	-	-
en1	1500	192.168.0	192.168.0.107	1646278	-	2156147	-	-
en2	1500	<Link#6>	d2:00:17:e4:d0:a0	0	0	0	0	0
fw0	4078	<Link#7>	40:6c:8f:ff:fe:7e:4d:0a	0	0	0	0	0
p2p0	2304	<Link#8>	02:56:81:97:14:31	0	0	0	0	0
bridg	1500	<Link#9>	42:6c:8f:32:5c:00	0	0	1	0	0
utun0	1380	<Link#10>		0	0	4	0	0
utun0	1380	fe80::415d:	fe80:a::415d:c924	0	-	4	-	-
utun0	1380	fdd6:d35e:5	fdd6:d35e:5451:e5	0	-	4	-	-

---

### 9.9.1 Tabela de Roteamento

Para visualizar a tabela de roteamento de um host podemos utilizar também o comando **netstat**. A Listagem 9.27 apresenta o comando.

**Listagem 9.27: Visualizando Tabela de Roteamento com netstat**

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ netstat -rn
Routing tables

Internet:
Destination      Gateway          Flags     Refs      Use   Netif Expire
default          192.168.0.1    UGSc        53       146    en1
127              127.0.0.1     UCS         0        0     lo0
127.0.0.1        127.0.0.1     UH          4       968    lo0
169.254          link#5       UCS         0        0    en1
192.168.0        link#5       UCS         4        0    en1
192.168.0.1/32   link#5       UCS         1        0    en1
192.168.0.1      1c:7e:e5:46:92:e7 UHLWIir   54      4585   en1    1173
192.168.0.103    7c:d1:c3:77:8:28 UHLWI      0       39    en1    1017
192.168.0.107/32 link#5       UCS         1        0    en1
192.168.0.107    70:56:81:97:14:31 UHLWI      0       4     lo0
192.168.0.108    a8:92:2c:ca:c2:d8 UHLWI      0       1    en1    886
192.168.0.109    1c:65:9d:82:e2:cd UHLWI      0      1087   en1    1143
192.168.0.150    70:62:b8:bf:fd:50 UHLWI      0      1366   en1    351

Internet6:
Destination          Gateway          Flags
Netif   Expire
default            fe80::7262:b8ff:febf:fd50%en1  UGcI
      en1
::1                ::1               UHL
      lo0
fdd6:d35e:5451:e50::/64   fe80::415d:c924:8387:28c1%utun0  Uc
      utun0
fdd6:d35e:5451:e50:415d:c924:8387:28c1  link#10             UHL
      lo0
fe80::%lo0/64        fe80::1%lo0           UcI
      lo0
fe80::1%lo0          link#1               UHLI
      lo0
fe80::%en1/64        link#5               UCI
      en1
fe80::7256:81ff:fe97:1431%en1    70:56:81:97:14:31        UHLI
      lo0
fe80::7262:b8ff:febf:fd50%en1    70:62:b8:bf:fd:50        UHLWIir
      en1
fe80::%utun0/64        fe80::415d:c924:8387:28c1%utun0  UcI
      utun0
fe80::415d:c924:8387:28c1%utun0  link#10             UHLI
      lo0
ff01::%lo0/32          ::1               UmCI
      lo0
ff01::%en1/32          link#5               UmCI
      en1
ff01::%utun0/32        fe80::415d:c924:8387:28c1%utun0  UmCI
      utun0
ff02::%lo0/32          ::1               UmCI
      lo0
ff02::%en1/32          link#5               UmCI
      en1
```

```
ff02::%utun0/32          fe80::415d:c924:8387:28c1%utun0 UmCI  
utun0
```

---

## 9.10 Comando nettop

O comando **nettop** [12] pode ser utilizado para monitorar o tráfego de rede. O comando funciona de forma muito parecido com o comando top (monitora processos). A Figura 9.1 apresenta o resultado do comando. Para sair da tela de comando digite q.

		interface	state
kernel_task.0	tcp4 127.0.0.1:1023<-->localhost:49153	lo0	Established
UserEventAgent.46	udp4 *:*<->*:*		
	udp4 *:*<->*:*		
syslogd.47	udp4 *:57001<->*:*	lo0	
configd.55	udp4 *:*<->*:*		
	udp4 *:*<->*:*		
airportd.62	udp4 *:*<->*:*		
	udp4 *:*<->*:*		
mtmfs.76	tcp4 127.0.0.1:49153<-->localhost:1023	lo0	Established
	tcp4 127.0.0.1:49153<-->*:*	lo0	Listen
	tcp4 127.0.0.1:49152<-->*:*	lo0	Listen
apsd.81	tcp4 192.168.0.107:64433<-->17.143.164.36:5223	en1	Established
	tcp4 192.168.0.107:56282<-->17.110.227.102:5223	en1	Established
locationd.87	udp4 *:*<->*:*		
blued.90			

Figura 9.1: Comando nettop

## 9.11 Network Mapper

O comando **nmap** é uma ferramenta excelente para fazer varreduras em redes de computadores.

### 9.11.1 Instalação

O arquivo binário pode ser obtido em [11]. O processo de instalação é semelhante a qualquer aplicativo do **MAC OS X**.

### 9.11.2 Analisando portas abertas

O comando da Listagem 9.28 apresenta a versão básica do comando para listar as portas abertas de um determinado domínio.

#### Listagem 9.28: Verificando Portas Abertas

```
musashi-2: alessandrovivas$ nmap www.ufvjm.edu.br
```

```
Starting Nmap 7.00 ( https://nmap.org ) at 2015-11-26 17:12 BRST
Nmap scan report for www.ufvjm.edu.br (200.131.252.28)
Host is up (0.24s latency).
Not shown: 996 filtered ports
PORT      STATE    SERVICE
80/tcp    open     http
443/tcp   open     https
4321/tcp  open     rwhois
8080/tcp  closed   http-proxy

Nmap done: 1 IP address (1 host up) scanned in 16.39 seconds
```

---

### 9.11.3 Comando nmap com opção de mais informações

Se quiser mais informações sobre o procedimento utilize a opção **-v** como na Listagem 9.29.

#### Listagem 9.29: Comando nmap com opção -v

```
musashi-2: alessandrovivas$ nmap -v www.ufvjm.edu.br

Starting Nmap 7.00 ( https://nmap.org ) at 2015-11-26 17:13 BRST
Initiating Ping Scan at 17:13
Scanning www.ufvjm.edu.br (200.131.252.28) [2 ports]
Completed Ping Scan at 17:13, 0.18s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 17:13
Completed Parallel DNS resolution of 1 host. at 17:13, 0.00s elapsed
Initiating Connect Scan at 17:13
Scanning www.ufvjm.edu.br (200.131.252.28) [1000 ports]
Discovered open port 80/tcp on 200.131.252.28
Discovered open port 443/tcp on 200.131.252.28
Completed Connect Scan at 17:14, 12.12s elapsed (1000 total ports)
Nmap scan report for www.ufvjm.edu.br (200.131.252.28)
Host is up (0.19s latency).
Not shown: 997 filtered ports
PORT      STATE    SERVICE
80/tcp    open     http
443/tcp   open     https
8080/tcp  closed   http-proxy

Read data files from: /usr/local/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 12.41 seconds
```

---

### 9.11.4 Rastreando Múltiplos Hosts

Para rastrear múltiplos hosts basta passar os endereços de IPs desejados. A Listagem 9.30 ilustra o procedimento para dois hosts

#### Listagem 9.30: Rastreando Múltiplos Hosts.numbers

```
musashi-2: alessandrovivas$ nmap 192.168.0.1 192.168.0.107

Starting Nmap 7.00 ( https://nmap.org ) at 2015-11-26 17:15 BRST
Nmap scan report for 192.168.0.1
Host is up (0.0100s latency).
Not shown: 998 closed ports
PORT      STATE    SERVICE
53/tcp    open     domain
980/tcp   open     http
```

```
Nmap scan report for 192.168.0.107
Host is up (0.00037s latency).
Not shown: 969 closed ports, 30 filtered ports
PORT      STATE SERVICE
153689/tcp open  rendezvous

Nmap done: 2 IP addresses (2 hosts up) scanned in 6.06 seconds
```

---

É possível rastrear múltiplos domínios como na Listagem 9.31

#### Listagem 9.31: Rastreando Múltiplos Domínios

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ nmap www.mit.edu www.ufvjm.edu.br

Starting Nmap 7.00 ( https://nmap.org ) at 2015-11-26 17:16 BRST
Nmap scan report for www.mit.edu (23.44.182.105)
Host is up (0.047s latency).
Other addresses for www.mit.edu (not scanned): 2600:1419:15:182::255e
2600:1419:15:186::255e
rDNS record for 23.44.182.105: a23-44-182-105.deploy.static.akamaitechnologies.com
Not shown: 998 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https

Nmap scan report for www.ufvjm.edu.br (200.131.252.28)
Host is up (0.19s latency).
Not shown: 996 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https
4321/tcp  open  rwhois
8080/tcp  closed http-proxy

Nmap done: 2 IP addresses (2 hosts up) scanned in 127.99 seconds
```

---

Outra opção é rastrear uma rede completa como na Listagem 9.32.

#### Listagem 9.32: Rastreando uma Sub-rede

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ nmap 192.168.0./*

Starting Nmap 7.00 ( https://nmap.org ) at 2015-11-26 17:19 BRST
Nmap scan report for 192.168.0.1
Host is up (0.0043s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http

Nmap scan report for 192.168.0.103
Host is up (0.011s latency).
All 1000 scanned ports on 192.168.0.103 are closed

Nmap scan report for 192.168.0.107
Host is up (0.00011s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
3689/tcp  open  rendezvous

Nmap scan report for 192.168.0.108
Host is up (0.027s latency).
```

```
All 1000 scanned ports on 192.168.0.108 are closed

Nmap scan report for 192.168.0.150
Host is up (0.0097s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
23/tcp    open  telnet
80/tcp    open  http
5431/tcp  open  park-agent

Nmap done: 256 IP addresses (5 hosts up) scanned in 13.60 seconds
```

---

## 9.12 Comando route

O comando **route** possibilita a manipulação de rotas de roteamento. Para adicionar uma nova rota na tabela de roteamento utilize o comando da Listagem 9.33 [7].

### Listagem 9.33: Adicionando uma rota

```
musashi-2:~ alessandrovivas$ sudo route -n add 172.18.0.0/23 192.168.10.1
add net 172.18.0.0: gateway 192.168.10.1
```

Como pode ser visualizado na Listagem 9.34 a roda foi adicionada corretamente.

### Listagem 9.34: Visualizando a Tabela de Roteamento

```
musashi-2:~ alessandrovivas$ netstat -rn
Internet:
Destination      Gateway          Flags       Refs      Use     Netif Expire
default          192.168.0.1    UGSc           40        0      en1
127              127.0.0.1     UCS            0        0      lo0
127.0.0.1        127.0.0.1     UH             3      1446      lo0
169.254          link#5        UCS            0        0      en1
172.18/23         192.168.10.1  UGSc           0        0      en1
```

Para deletar a rota utilize o comando da Listagem 9.35 a roda foi adicionada corretamente.

### Listagem 9.35: Visualizando a Tabela de Roteamento

```
musashi-2:~ alessandrovivas$ sudo route -n delete 172.18.0.0/23 192.168.10.1
delete net 172.18.0.0: gateway 192.168.10.1
```

Na Listagem 9.36 pode ser visto que a rota foi apagada.

### Listagem 9.36: Visualizando a Tabela de Roteamento

```
musashi-2:~ alessandrovivas$ netstat -rn
Routing tables

Internet:
Destination      Gateway          Flags       Refs      Use     Netif Expire
default          192.168.0.1    UGSc           20        0      en1
127              127.0.0.1     UCS            0        0      lo0
127.0.0.1        127.0.0.1     UH             3      1446      lo0
```

```

169.254      link#5        UCS          0      0      en1
192.168.0     link#5        UCS          2      0      en1
192.168.0.1/32 link#5        UCS          1      0      en1
192.168.0.1    1c:7e:e5:46:92:e7 UHLWIir    22     1081   en1    1199
192.168.0.107/32 link#5        UCS          1      0      en1
192.168.0.107  70:56:81:97:14:31 UHLWI       0      4      lo0
192.168.0.109  70:62:b8:bf:fd:50 UHLWI       0      0      en1    1189
192.168.0.255  ff:ff:ff:ff:ff:ff UHLWbI      0      2      en1
...

```

---

## 9.13 Comando telnet

O comando **telnet** foi muito utilizado como protocolo de acesso remoto. Aos poucos foi substituído pelo comando **ssh** devido a problemas de segurança.

### 9.13.1 Acessando Servidor Web via Telnet

Na Listagem 9.37 apresenta o comando para conectar ao servidor desejado.

#### Listagem 9.37: Uso do Telnet

```
$ telnet www.vivas.eng.br 80
```

Você irá receber a seguinte resposta do servidor como na Listagem 9.38.

#### Listagem 9.38: Resposta do Servidor

```

Trying 208.115.217.250...
Connected to vivas.eng.br.
Escape character is '^]'.

```

Digite os comandos seguintes da Listagem 9.39 e termine pressionando duas vezes enter.

#### Listagem 9.39: Acessando com Telnet o Servidor Web

```

GET / HTTP/1.1
Host: vivas.eng.br

```

Como resposta o servidor enviará informações do protocolo e enviará a página desejada como na Listagem 9.40.

#### Listagem 9.40: Resposta do Servidor

```

musashi-2:LivroMac-1aEdicao alessandrovivas$ telnet www.vivas.eng.br 80
Trying 158.69.55.227...
Connected to vivas.eng.br.
Escape character is '^>'.
GET / HTTP/1.1
Host : vivas.eng.br

HTTP/1.1 200 OK
Date: Fri, 27 Nov 2015 18:13:24 GMT
Server: Apache
Last-Modified: Tue, 03 Nov 2015 22:00:06 GMT
Accept-Ranges: bytes
Content-Length: 5885

```

Connection: close  
Content-Type: text/html

```
<!DOCTYPE HTML>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8" />
    <title>Alessandro Vivas Andrade</title>
    <!-- Aqui chamamos o nosso arquivo css externo -->
    <link rel="stylesheet" type="text/css" href="style.css" />
    <!--[if lte IE 8]>
        <script src="https://cdnjs.cloudflare.com/ajax/libs/html5shiv/3.7.3/html5shiv.js"></script>
    <![endif]-->
</head>
<body>
<nav>
    <ul>
        <li><a href="#">Home</a></li>
        <li><a href="#">Pesquisa</a>
            <div>
                <ul>
                    <li><a href="#">Áreas de Interesse</a></li>
                    <li><a href="#">Grupo de Pesquisa</a></li>
                    <li><a href="#">Artigos Publicados</a></li>
                    <li><a href="#">Projetos de Pesquisa</a></li>
                    <li><a href="#">Patentes</a></li>
                    <li><a href="#">Softwares</a></li>
                </ul>
            </div>
        </li>
        <li><a href="#">Mestrando</a>
            <div>
                <ul>
                    <li><a href="#">Áreas de Interesse</a></li>
                    <li><a href="#">Alunos de Mestrando</a></li>
                    <li><a href="#">Disciplinas</a></li>
                    <li><a href="#">Bancas de Mestrando</a></li>
                    <li><a href="#">Dissertações</a></li>
                    <li><a href="#">Quer Trabalhar Comigo?</a></li>
                </ul>
            </div>
        </li>
    </ul>
</nav>
<div>
    <ul>
        <li><a href="#">Disciplinas Graduação <span class="dropdown"></span></a>
            <div>
                <ul>
                    <li><a href="#">Redes de Computadores 2</a></li>
                    <li><a href="#">Sistemas Distribuídos</a></li>
                    <li><a href="#">Sistemas Operacionais</a></li>
                        <ul>
                            <li><a href="#">Tópicos Especiais: Linux</a></li>
                            <li><a href="#">Inteligência Artificial</a></li>
                            <li><a href="#">Pesquisa Operacional</a></li>
                            <li><a href="#">Sistemas de Computação</a></li>
                            <li><a href="#">Organização e Arquitetura de Computadores</a></li>
                        </ul>
                    <li><a href="#">Linguagens de Programação</a></li>
                </ul>
            </div>
        </li>
    </ul>
</div>
```

```

        </ul>
    </div>
</li>
<li>
    <a href=          >Disciplinas Pós-Graduação <span class=
        <div>
    <ul>

        <li><a href=      >Pesquisa Orientada</a></li>
        <li><a href=      >Tecnologias de Educação Aplicadas À Educa
            ção</a></li>
        <li><a href=      >Processos Avaliativos e Tecnologias
            Aplicados À Educação Superior</a></li>

    </ul>
    </div> </li>

    <li><a href=      >TCC</a>
        <div>
    <ul>
        <li><a href=      >Áreas de Trabalho</a></li>
        <li><a href=      >Monografias Orientadas</a></li>
        <li><a href=      >Quer Trabalhar Comigo?</a></li>
    </ul>
    </div>

    </li>

<li>
    <a href=          >Cursos <span class=      ></span></a>
    <div>
        <ul>
            <li><a href=      >Linux Comandos Básicos</a></li>
            <li><a href=      >Introdução ao Sistema Operacional Linux</a
                ></li>

        </ul>
    </div>
</li>

    <li>
        <a href=          >Palestras <span class=      ></span></a>
        <div>
            <ul>
                <li><a href=      >Computação Bio-inspirada</a></li>

            </ul>
        </div>
    </li>

    <li>
        <a href=          >Livros <span class=      ></span></a>
        <div>
            <ul>
                <li><a href=      >Latex</a></li>

            </ul>
        </div>
    </li>

</ul>

```

```

        </div>

    </li>

    <li>
        <a href=
            >Bancas <span class=           ></span></a>
        </li>

        <li><a href=                  >Interesses Pessoais</a>
            <div>
                <ul>
                    <li><a href=          >Observação de Aves</a></li>
                    <li><a href=          >Fotografia</a></li>
                    <li><a href=          >Mountain Bike</a></li>
                    <li><a href=          >Música</a></li>
                    <li><a href=          >Corrida</a></li>
                    <li><a href=          >Viagens</a></li>
                </ul>
            </div>
        </li>

        </ul>
    </nav>

    <img src=vivas.jpg border=      width=           height=           ALIGN=MIDDLE> <br>

<H2>Alessandro Vivas Andrade</H2>
Departamento de Computação<br>
Universidade Federal dos Vales do Jequitinonha e Mucuri<br>
Diamantina, MG <br>
<em>prof.alessandrovivas@gmail.com</em><br>
<em>Skype: </em>alessandrovivas <br>
<em>Lattes: </em>http://lattes.cnpq.br/5412055666902423 <br>

<H3>Formação</H3>
<ul>
    <li> Doutorado em Engenharia Elétrica - UFMG
    <li> Mestrado em Engenharia Elétrica - UFMG
    <li> Graduação em Engenharia Elétrica - UFMG
    <li> Técnico em Eletrônica - Escola Técnica Franciso Moreira da Costa
</ul>

<H3>Experiência Profissional</H3>
<ul>
    <li> Professor do Curso de Sistemas de Informação - UFVJM
    <li> Professor do Curso de Computação - UNIBH
    <li> Engenheiro de Telecomunicações - TIM
</ul>

```

```
<H3>Experiênci Administrativa</H3>
<ul>
<li> Chefe de Departamento
<li> Vice-Coordenação de Curso
</ul>

<H3>Endereço Profissional</H3>
UFVJM – Campus 2 <br>
Sala 01 – Prédio de Sistemas de Informação <br>
Rodovia MGT 367 – km 583 – número 5000 <br>
Alto da Jacuba – CEP 39100-000 <br>
Diamantina, Minas Gerais <br>
</body>
</html>
Connection closed by foreign host.
```

---

## 9.14 Acesso Remoto com ssh

O comando **ssh** permite o acesso remoto a um servidor. O primeiro passo é habilitar o serviço de ssh. Entre na interface gráfica e procure "Preferências de Sistema", depois Compartilhamento e clique em "login remoto"[13].

### 9.14.1 Acesso Remoto

Para acessar remotamente um servidor basta você fazer o procedimento da Listagem 9.41. Onde **alessandrovivas** é o usuário e o endereço IP do servidor é 192.168.0.1.

#### Listagem 9.41: Utilizando o ssh

```
$ ssh alessandrovivas@192.168.0.1
```

---

## 9.15 Copiando Arquivos com scp

O ssh permite também que você copie um arquivo de um computador remoto para outro computador remoto. Neste exemplo vou copiar o arquivo *teste.txt* (que está no diretório */home/vivas*) que está no computador 192.168.0.104 para o meu computador para o diretório */users/alessandrovivas*. Repare que você vai digitar a senha do seu computador remoto e não do computador que você está logado . Para fazer esta tarefa utilize o comando **scp** e o código está na Listagem 9.42.

#### Listagem 9.42: Copiando Arquivo em Servidor Remoto

```
$ scp vivas@192.168.0.104:/home/vivas/teste.txt /users/alessandrovivas
vivas@192.168.0.104's password:
teste.txt 100% 438 0.4KB/s 00:00
```

---

## 9.16 Copiando um Diretório de um Servidor Remoto

Imagine que você criou um diretório em um servidor remoto. Vamos supor que o diretório tem o nome de ubuntu (`/home/vivas/ubuntu`). Para copiar o diretório inteiro, todos os arquivos, e criar a mesma estrutura no seu computador basta usar o comando `scp`. O código está apresentado na Listagem 9.43 e vamos utilizar o comando `scp`.

#### Listagem 9.43: Copiando um Diretório de um Servidor Remoto

```
$ scp -r vivas@192.168.0.104:/home/vivas/ubuntu /users/alessandrovivas/
vivas@192.168.0.104's password:
arquivo2      100%    0     0.0KB/s   00:00
arquivo1      100%    0     0.0KB/s   00:00
arquivo3      100%    0     0.0KB/s   00:00
```

---

## 9.17 Comando tcpdump

O comando **tcpdump** é utilizado para obter informações de suas conexões de rede e pode atuar como um sniffer. Uma boa referência de variações do `tcpdump` é o sítio A `tcpdump Primer with Examples` [10].

### 9.17.1 Listando Todas as Interfaces de Rede

Para listar as interfaces de rede que ele pode escutar utilize a Listagem 9.44.

#### Listagem 9.44: Interfaces que podem ser utilizadas com tcpdump

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ tcpdump -D
1.en0
2/fw0
3.bridge0
4.utun0
5.en1
6.en2
7.p2p0
8.lo0 [Loopback]
```

---

### 9.17.2 Realizando Captura de Pacotes da Interface Sem Fio

Para realizar a captura de pacotes da interface de rede utilize a Listagem 9.45. Para sair digite `<Ctrl+C>`.

#### Listagem 9.45: Capturando Pacotes da Interface de Rede Sem Fio

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ tcpdump -i en1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on en1, link-type EN10MB (Ethernet), capture size 262144 bytes
16:39:43.843607 IP 192.168.0.107.63189 > ec2-54-229-254-54.eu-west-1.compute.amazonaws.com.https: Flags [.], ack 4114535125, win 4096, length 0
16:39:44.217087 IP ec2-54-229-254-54.eu-west-1.compute.amazonaws.com.https >
192.168.0.107.63189: Flags [.], ack 1, win 83, options [nop,nop,TS val 123269221 ecr 285851695], length 0
16:39:44.311854 ARP, Request who-has 192.168.0.199 tell 192.168.0.1, length 46
16:39:44.374463 IP 192.168.0.107.61631 > 192.168.0.1.domain: 42742+ PTR? 107.0.168.192.in-addr.arpa. (44)
16:39:44.432987 IP 192.168.0.1.domain > 192.168.0.107.61631: 42742 NXDomain 0/0/0 (44)
16:39:44.435481 IP 192.168.0.107.55786 > 192.168.0.1.domain: 9359+ PTR? 54.254.229.54.in-addr.arpa. (44)
```

`^C`

```
41 packets captured
55 packets received by filter
0 packets dropped by kernel
```

---

### 9.17.3 Capturando Pacotes de um Host Específico

Para realizar a captura de pacotes de um determinado host utilize a Listagem 9.46. Para sair digite <Ctrl+C>.

#### Listagem 9.46: Capturando Pacotes da Interface de um Host Específico

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ sudo tcpdump host 192.168.0.1
Password:
tcpdump: data link type PKTAP
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on pktap, link-type PKTAP (Packet Tap), capture size 262144 bytes
16:42:21.613366 IP 192.168.0.107.mdns > 192.168.0.1.nat-pmp: [|domain]
16:42:21.613627 IP 192.168.0.107.63491 > 192.168.0.1.65535: Flags [S], seq 1745139063,
    win 65535, options [mss 1460,nop,wscale 5,nop,nop,TS val 286053897 ecr 0,sackOK,eol],
    length 0
16:42:21.615241 IP 192.168.0.1.65535 > 192.168.0.107.63491: Flags [S.], seq 1493936435,
    ack 1745139064, win 5792, options [mss 1460,sackOK,TS val 285594470 ecr 286053897,nop
    ,wscale 1], length 0
16:42:21.615285 IP 192.168.0.107.63491 > 192.168.0.1.65535: Flags [.], ack 1, win 4117,
    options [nop,nop,TS val 286053898 ecr 285594470], length 0
16:42:21.615332 IP 192.168.0.107.63491 > 192.168.0.1.65535: Flags [P.], seq 1:626, ack 1,
    win 4117, options [nop,nop,TS val 286053898 ecr 285594470], length 625
16:42:21.616752 IP 192.168.0.1.65535 > 192.168.0.107.63491: Flags [.], ack 626, win 3521,
    options [nop,nop,TS val 285594471 ecr 286053898], length 0
16:42:21.617416 IP 192.168.0.1.65535 > 192.168.0.107.63491: Flags [P.], seq 1:520, ack
    626, win 3521, options [nop,nop,TS val 285594472 ecr 286053898], length 519
16:42:21.617441 IP 192.168.0.107.63491 > 192.168.0.1.65535: Flags [.], ack 520, win 4101,
    options [nop,nop,TS val 286053899 ecr 285594472], length 0
16:42:21.617810 IP 192.168.0.1.65535 > 192.168.0.107.63491: Flags [F.], seq 520, ack 626,
    win 3521, options [nop,nop,TS val 285594472 ecr 286053898], length 0
16:42:21.617852 IP 192.168.0.107.63491 > 192.168.0.1.65535: Flags [.], ack 521, win 4101,
    options [nop,nop,TS val 286053900 ecr 285594472], length 0
16:42:21.618022 IP 192.168.0.107.63491 > 192.168.0.1.65535: Flags [F.], seq 626, ack 521,
    win 4101, options [nop,nop,TS val 286053900 ecr 285594472], length 0
16:42:21.618795 IP 192.168.0.1.65535 > 192.168.0.107.63491: Flags [.], ack 627, win 3521,
    options [nop,nop,TS val 285594473 ecr 286053900], length 0
16:42:22.105684 IP 192.168.0.107.59875 > 192.168.0.1.domain: 54439+ PTR? 107.0.168.192.in
    -addr.arpa. (44)
16:42:22.235619 IP 192.168.0.1.domain > 192.168.0.107.59875: 54439 NXDomain 0/0/0 (44)
16:42:22.237304 IP 192.168.0.107.60274 > 192.168.0.1.domain: 21135+ PTR? 1.0.168.192.in-
    addr.arpa. (42)
16:42:22.365378 IP 192.168.0.1.domain > 192.168.0.107.60274: 21135 NXDomain 0/0/0 (42)
^C
16 packets captured
33 packets received by filter
0 packets dropped by kernel
```

---

### 9.17.4 Capturando Pacotes de uma Rede

Para realizar a captura de pacotes de uma determinada rede utilize a Listagem 9.47. Para sair digite <Ctrl+C>.

#### Listagem 9.47: Capturando Pacotes da Interface de um Host Específico

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ sudo tcpdump net 192.168.0.0/24
tcpdump: data link type PKTAP
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on ptkap, link-type PKTAP (Packet Tap), capture size 262144 bytes
16:45:01.948116 IP 6.97.a86c.ip4.static.sl-reverse.com.https > 192.168.0.107.63394: Flags
    [.], ack 510170416, win 144, options [nop,nop,TS val 952219009 ecr 286213501],
    length 0
16:45:02.271947 ARP, Request who-has 192.168.0.162 tell 192.168.0.1, length 46
16:45:02.849132 IP 192.168.0.107.60455 > 192.168.0.1.domain: 17823+ PTR? 6.151.168.108.in
    -addr.arpa. (44)
16:45:02.951566 IP 192.168.0.1.domain > 192.168.0.107.60455: 17823 1/0/0 PTR 6.97.a86c.
    ip4.static.sl-reverse.com. (93)
16:45:02.953156 IP 192.168.0.107.55179 > 192.168.0.1.domain: 58164+ PTR? 107.0.168.192.in
    -addr.arpa. (44)
16:45:02.987690 IP 192.168.0.1.domain > 192.168.0.107.55179: 58164 NXDomain 0/0/0 (44)
16:45:02.989301 IP 192.168.0.107.54736 > 192.168.0.1.domain: 57012+ PTR? 162.0.168.192.in
    -addr.arpa. (44)
16:45:03.025017 IP 192.168.0.1.domain > 192.168.0.107.54736: 57012 NXDomain 0/0/0 (44)
16:45:03.026018 IP 192.168.0.107.60563 > 192.168.0.1.domain: 30430+ PTR? 1.0.168.192.in-
    addr.arpa. (42)
16:45:03.048027 IP 192.168.0.1.domain > 192.168.0.107.60563: 30430 NXDomain 0/0/0 (42)
16:45:03.193311 ARP, Request who-has 192.168.0.159 tell 192.168.0.1, length 46
16:45:03.193520 ARP, Request who-has 192.168.0.158 tell 192.168.0.1, length 46
16:45:03.194013 ARP, Request who-has 192.168.0.157 tell 192.168.0.1, length 46
16:45:03.612324 IP 192.168.0.101.mdns > 224.0.0.251.mdns: 0*- [0q] 1/0/1 (Cache flush)
    TXT
    (94)
16:45:04.051076 IP 192.168.0.107.59001 > 192.168.0.1.domain: 6619+ PTR? 159.0.168.192.in-
    addr.arpa. (44)
16:45:04.091695 IP 192.168.0.1.domain > 192.168.0.107.59001: 6619 NXDomain 0/0/0 (44)
16:45:04.092883 IP 192.168.0.107.56539 > 192.168.0.1.domain: 6046+ PTR? 158.0.168.192.in-
    addr.arpa. (44)
16:45:04.146921 IP 192.168.0.1.domain > 192.168.0.107.56539: 6046 NXDomain 0/0/0 (44)
16:45:04.147965 IP 192.168.0.107.62287 > 192.168.0.1.domain: 17883+ PTR? 157.0.168.192.in
    -addr.arpa. (44)
16:45:04.215769 IP 192.168.0.1.domain > 192.168.0.107.62287: 17883 NXDomain 0/0/0 (44)
16:45:04.217098 IP 192.168.0.107.52822 > 192.168.0.1.domain: 15719+ PTR? 101.0.168.192.in
    -addr.arpa. (44)
16:45:04.225365 ARP, Request who-has 192.168.0.156 tell 192.168.0.1, length 46
16:45:04.225645 ARP, Request who-has 192.168.0.155 tell 192.168.0.1, length 46
16:45:04.226053 ARP, Request who-has 192.168.0.154 tell 192.168.0.1, length 46
16:45:04.308067 IP 192.168.0.107.61105 > edge-star-shv-01-mial.facebook.com.https: Flags
    [.], ack 2003894170, win 4096, length 0
16:45:04.309128 IP 192.168.0.1.domain > 192.168.0.107.52822: 15719 NXDomain 0/0/0 (44)
16:45:04.310632 IP 192.168.0.107.54625 > 192.168.0.1.domain: 63501+ PTR? 251.0.0.224.in-
    addr.arpa. (42)
16:45:04.364532 IP 192.168.0.1.domain > 192.168.0.107.54625: 63501 NXDomain 0/1/0 (99)
16:45:04.402776 IP 192.168.0.107.63407 > ec2-54-243-116-238.compute-1.amazonaws.com.https
    : Flags [P.], seq 2215673513:2215674218, ack 3289922633, win 4096, options [nop,nop,
    TS val 286216126 ecr 1295397623], length 705
16:45:04.463013 IP edge-star-shv-01-mial.facebook.com.https > 192.168.0.107.61105: Flags
    [.], ack 1, win 2043, options [nop,nop,TS val 541056159 ecr 286171032], length 0
16:45:04.598765 IP ec2-54-243-116-238.compute-1.amazonaws.com.https >
    192.168.0.107.63407: Flags [P.], seq 1:243, ack 705, win 136, options [nop,nop,TS val
    1295401376 ecr 286216126], length 242
16:45:04.598850 IP 192.168.0.107.63407 > ec2-54-243-116-238.compute-1.amazonaws.com.https
    : Flags [.], ack 243, win 4088, options [nop,nop,TS val 286216321 ecr 1295401376],
    length 0
16:45:04.839944 IP 192.168.0.109.62307 > 239.255.255.250.ssdp: UDP, length 133
16:45:05.245413 ARP, Request who-has 192.168.0.153 tell 192.168.0.1, length 46
16:45:05.245888 ARP, Request who-has 192.168.0.152 tell 192.168.0.1, length 46
16:45:05.246153 ARP, Request who-has 192.168.0.151 tell 192.168.0.1, length 46
^C
36 packets captured
```

```
59 packets received by filter
0 packets dropped by kernel
```

---

## 9.18 Baixando Sites com wget

Para baixar um site inteiro podemos utilizar o comando **wget**. A sintaxe é apresentada na Listagem 9.48 e para interromper digite <Ctrl+C>.

**Listagem 9.48:** Baixando Sites com wget

```
$ wget --recursive www.vivas.eng.br
--2013-12-18 07:49:33--  http://www.vivas.eng.br/
Resolvendo www.vivas.eng.br (www.vivas.eng.br)... 208.115.217.250
Conectando-se a www.vivas.eng.br (www.vivas.eng.br)|208.115.217.250|:80... conectado.
A requisição HTTP foi enviada, aguardando resposta... 200 OK
Tamanho: 7032 (6,9K) [text/html]
Salvando em: www.vivas.eng.br/index.html

100%[=====] 7.032
    11,3K/s   em 0,6s

2013-12-18 07:49:35 (11,3 KB/s) - www.vivas.eng.br/index.html salvo [7032/7032]
```

---

Para baixar um arquivo com wget basta passar como argumento o endereço de onde o arquivo se encontra.



# Compactação e Backup

*Eu acredito que talvez há no mercado espaço para uns cinco computadores..*

Thomas Watson, Chairman da IBM, em 1943.

## Sumário

<b>10.1 Compactando e Descompactando com zip . . . . .</b>	<b>108</b>
<b>10.2 Compactando e Descompactando com .tar . . . . .</b>	<b>109</b>
<b>10.3 Compactando e Descompactando com .tar.gz . . . . .</b>	<b>109</b>
<b>10.4 Compactando e Descompactando com .tar.bz2 . . . . .</b>	<b>109</b>
<b>10.5 Manipulando Imagens .dmg . . . . .</b>	<b>110</b>

## 10.1 Compactando e Descompactando com zip

### 10.1.1 Compactando com zip

O comando **zip** é utilizado para compactação de arquivos no formato **.zip**. A Listagem 10.1 apresenta o procedimento para compactar um diretório.

#### Listagem 10.1: Compactando um Diretório com zip

```
musashi-2:Livros alessandrovivas$ zip -r LivroMac-1aEdicao.zip LivroMac-1aEdicao/
  adding: LivroMac-1aEdicao/ (stored 0%)
  adding: LivroMac-1aEdicao/.comandos.txt.swp (deflated 67%)
  adding: LivroMac-1aEdicao/.DS_Store (deflated 91%)
  adding: LivroMac-1aEdicao/act.00 (deflated 57%)
  adding: LivroMac-1aEdicao/act.01 (deflated 57%)
  adding: LivroMac-1aEdicao/act.02 (deflated 57%)
  adding: LivroMac-1aEdicao/act.03 (deflated 57%)
  adding: LivroMac-1aEdicao/act.04 (deflated 57%)
...
...
```

---

Para compactar todos arquivos de um diretório utilize a Listagem 10.2.

#### Listagem 10.2: Compactando arquivos com zip

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ zip arquivos.zip *.*
  adding: Caracara01.JPG (deflated 1%)
  adding: Caracara02.JPG (deflated 1%)
  adding: Caracara03.JPG (deflated 1%)
  adding: MaclaEdicao.aux (deflated 72%)
  adding: MaclaEdicao.bbl (deflated 57%)
  adding: MaclaEdicao.blg (deflated 58%)
...
...
```

---

### 10.1.2 Descompactando com zip

O comando **unzip** é utilizado para descompactação de arquivos no formato **.zip**. A Listagem 10.3 apresenta o procedimento para descompactar um arquivo.

#### Listagem 10.3: Descompactando um Diretório com unzip

```
musashi-2:Livros alessandrovivas$ unzip LivroMac-1aEdicao.zip
```

---

### 10.1.3 Informações sobre a compactação

Se quiser obter detalhes sobre a compactação utilize o comando **zipinfo** como na Listagem 10.4.

#### Listagem 10.4: Verificando Informações com zipinfo

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ zipinfo arquivos.zip
Archive: arquivos.zip  20092480 bytes  118 files
-rwxr-xr-x 3.0 unx 1362100 bx defN 16-Mar-13 09:48 Caracara01.JPG
-rwxr-xr-x 3.0 unx 1320772 bx defN 16-Mar-13 09:48 Caracara02.JPG
-rwxr-xr-x 3.0 unx 1357615 bx defN 16-Mar-13 09:49 Caracara03.JPG
-rw-r--r-- 3.0 unx     3356 tx defN 29-Nov-15 08:29 MaclaEdicao.aux
...
118 files, 22378701 bytes uncompressed, 20074190 bytes compressed: 10.3%
```

## 10.2 Compactando e Descompactando com .tar

### 10.2.1 Compactando com tar (gerando arquivos .tar)

Outra forma de compactação é utilizar o comando **tar**. Você pode gerar um comando **.tar** utilizando o comando da Listagem 10.6.

#### Listagem 10.5: Compactando um Diretório com tar

```
musashi-2:Livros alessandrovivas$ tar -cvf diretorio.tar Diretorio/
```

### 10.2.2 Descompactando com tar (arquivos .tar)

Para descompactar um arquivo **.tar** basta utilizar o comando da Listagem 10.6.

#### Listagem 10.6: Descompactando um arquivo com extensão tar

```
musashi-2:Livros alessandrovivas$ tar -xvf diretorio.tar
```

## 10.3 Compactando e Descompactando com .tar.gz

### 10.3.1 Compactando com tar e gzip (gerando arquivos .tar.gz)

Outra forma de compactação é utilizar o comando **tar**. Você pode gerar um arquivo **.tar.gz** utilizando o comando da Listagem 10.7.

#### Listagem 10.7: Compactando um Diretório com tar.gz

```
musashi-2:Livros alessandrovivas$ tar -zcvf diretorio.tar.gz Diretorio/
```

### 10.3.2 Descompactando com tar e gzip (arquivos .tar.gz)

Para descompactar um arquivo **.tar.gz** basta utilizar o comando da Listagem 10.8.

#### Listagem 10.8: Descompactando um arquivo com extensão tar.gz

```
musashi-2:Livros alessandrovivas$ tar -zxvf diretorio.tar
```

## 10.4 Compactando e Descompactando com .tar.bz2

### 10.4.1 Compactando com tar e gzip (gerando arquivos .tar.bz2)

Outra forma de compactação é utilizar o comando **tar**. Você pode gerar um arquivo **.tar.bz2** utilizando o comando da Listagem 10.9.

#### Listagem 10.9: Compactando um Diretório com tar.gz

```
musashi-2:Livros alessandrovivas$ tar -jcvf diretorio.tar.bz2 Diretorio/
```

## 10.4.2 Descompactando com tar e gzip (arquivos .tar.bz2)

Para descompactar um arquivo **.tar.bz2** basta utilizar o comando da Listagem 10.10.

### Listagem 10.10: Descompactando um arquivo com extensão tar.bz2

```
musashi-2:Livros alessandrovivas$ tar -jxvf diretorio.tar.bz2
```

Você pode também usar o comando **bzip2** para compactar diretamente arquivos. Vamos supor que quero compactar todos os arquivos com extensão pdf no diretório corrente, Listagem 10.11.

### Listagem 10.11: Compactando com bzip2

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ bzip2 *.pdf  
% verificando  
musashi-2:LivroMac-1aEdicao alessandrovivas$ ls *.bz2  
Mac1aEdicao.pdf.bz2      UmaAbordagemEstocastica.pdf.bz2
```

Para descompactar utilize o comando da Listagem 10.12.

### Listagem 10.12: Descompactando com bzip2

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ bzip2 -d *.bz2
```

## 10.5 Manipulando Imagens .dmg

Imagens são arquivos que contém a estrutura completa de um volume ou diretório. Este arquivo compactado quando descompactado apresenta toda a estrutura original do item compactado. A estrutura do item copiado é igual a cópia, bit a bit. Este tipo de compactação mantém também as informações de boot caso necessário.

### 10.5.1 Criando Imagens

Para criar uma imagem no formato **.dmg** deve-se utilizar o comando **hdiutil**. Caso tenha alguma dúvida consulte a referência [6]. Ele pode precisar de alguns dos seguintes argumentos:

- UDZO - nome da imagem
- UDRO - imagem para apenas leitura
- UDBZ - imagem com compressão
- URDW - imagem com permissão de leitura e escrita
- UDTO - imagem de DVD

Para criar a imagem utilize a Listagem 10.13, este comando irá compactar o diretório LivroMac-1aEdicao e gerar a imagem Livro.dmg.

### Listagem 10.13: Criando Imagem dmg

```
musashi-2:Livros alessandrovivas$ hdiutil create -format UDZO -srcfolder ./LivroMac-1  
aEdicao/ Livro.dmg  
.....  
created: /Users/alessandrovivas/Documents/Livros/Livro.dmg
```

Se quiser utilizar o arquivo é preciso montar a imagem como na Listagem 10.14.

#### Listagem 10.14: Criando Imagem dmg

```
musashi-2:Livros alessandrovivas$ hdiutil attach Livro.dmg
esperado    CRC32 $48CDC2FC
/dev/disk2          GUID_partition_scheme
/dev/disk2s1        Apple_HFS
                                /Volumes/LivroMac-1aEdicao
```

---

Se quiser desmontar o arquivo utilize a Listagem 10.15

#### Listagem 10.15: Ejetando Volume dmg

```
musashi-2:Livros alessandrovivas$ hdiutil eject /Volumes/LivroMac-1aEdicao/
unmounted.
ejected.
```

---

### 10.5.2 Duplicando uma Imagem

Se precisar duplicar uma imagem de um volume para outro volume, como por exemplo HD Externo, utilize o comando **asr**. A Listagem 10.16 apresenta o procedimento para o comando.

#### Listagem 10.16: Copiando Imagem com asr

```
sudo asr restore -source /Volumes/Vivas --target /Volumes/Backup -noprompt -noverify -
erase
```

---



# Comandos Úteis

*Não há motivo algum para que  
as pessoas queiram ter um  
computador em casa.*

Ken Olsen, presidente da Digital Equipment Corp, em 1977

## Sumário

11.1 Comando say . . . . .	114
11.2 Imprimir Metadados com mtl\$ . . . . .	114
11.3 Comando mdfind . . . . .	115
11.4 Comando units . . . . .	116
11.5 Comando yes . . . . .	117
11.6 Linguagem awk . . . . .	117

## 11.1 Comando say

O comando **say** pode ser utilizado para usar o terminal para vocalizar frases e arquivos inteiros. Uma boa referência para a configuração pode ser encontrada em um artigo da Tekrevue [19]. Para configurar o sistema para o idioma português selecione na interface gráfica Preferência de Sistema e depois Ditado e Fala.

### 11.1.1 Passando uma Frase

O comando **say** pode ser utilizado para falar uma frase específica como na Listagem 11.1. A sintaxe é bem simples basta passar como argumento o texto desejado.

#### Listagem 11.1: Frase específica com o comando say

```
musashi-2: alessandrovivas$ say alessandro vivas andrade
```

### 11.1.2 Passando um Arquivo como Argumento

Você pode passar também como um argumento um arquivo completo apenas utilizando a opção **-f**. A Listagem 11.2 apresenta o comando completo.

#### Listagem 11.2: Arquivo com o comando say

```
musashi-2: alessandrovivas$ say -f hamlet.txt
```

### 11.1.3 Passando o Locutor como Argumento

Uma opção para o comando **say** é passar o locutor como argumento. A Listagem 11.3 mostra como selecionar a locutora Luciana para falar o arquivo **hamlet.txt**.

#### Listagem 11.3: Selecionando locutor com o comando say

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ say -v Luciana -f hamletpt.txt
```

## 11.2 Imprimir Metadados com mtlst

O comando **mtls** é usado para imprimir metadados de arquivos. A Listagem 11.4 apresenta o resultado do comando em um arquivo do formato **png**.

#### Listagem 11.4: Lendo Metadados com mtlst

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ mdls LinkParaFirefox.png
_kMDItemUserID          = 501
_kTimeMachineIsCreationMarker = 1
_kTimeMachineNewestSnapshot = 4001-01-01 00:00:00 +0000
_kTimeMachineOldestSnapshot = 2015-11-30 18:36:21 +0000
_kMDItemAlternateNames   =
)
kMDItemBitsPerSample      = 32
kMDItemColorSpace         =
kMDItemContentCreationDate = 2013-12-18 09:03:50 +0000
kMDItemContentModificationDate = 2013-12-18 09:03:50 +0000
```

```

kMDItemContentType          =
kMDItemContentTypeTree      =
,
,
,
)

kMDItemDateAdded           = 2015-11-26 10:16:20 +0000
kMDItemDisplayName          =
kMDItemFSContentChangeDate = 2013-12-18 09:03:50 +0000
kMDItemFSCreationDate       = 2013-12-18 09:03:50 +0000
kMDItemFSCreatorCode        =
kMDItemFSFinderFlags        = 16
kMDItemFSHasCustomIcon      = (null)
kMDItemFSInvisible          = 0
kMDItemFSIsExtensionHidden  = 1
kMDItemFSIsStationery       = (null)
kMDItemFSLabel               = 0
kMDItemFSName                =
kMDItemFSNodeCount          = (null)
kMDItemFSOwnerGroupID        = 20
kMDItemFSOwnerUserID         = 501
kMDItemFSSize                = 53756
kMDItemFSTypeCode            =
kMDItemHasAlphaChannel      = 0
kMDItemKind                 =
kMDItemLogicalSize           = 53756
kMDItemOrientation           = 0
kMDItemPhysicalSize          = 57344
kMDItemPixelCount             = 393084
kMDItemPixelHeight            = 549
kMDItemPixelWidth              = 716
kMDItemProfileName           =
kMDItemResolutionHeightDPI   = 0
kMDItemResolutionWidthDPI    = 0

```

---

## 11.3 Comando mdfind

O comando **mdfind** pode ser utilizado para procurar arquivos através de seus atributos. Os atributos mais comuns são:

- kMDItemPath: Caminho Absoluto
- kMDItemFSName: nome
- kMDItemFSSize: tamanho em bytes
- kMDItemDisplayName: nome mostrado
- kMDItemKind: conteúdo
- kMDItemFSContentChangeDate: data de modificação
- kMDItemFSCreationDate: data de criação
- kMDItemLastUsedDate: última vez que foi aberto
- kMDItemDateAdded: data que foi adicionado
- kMDItemFinderComment: comentário do spotlight
- kMDItemTextContent: tipo de arquivo HTML, texto ou PDF
- kMDItemDurationSeconds: tempo de duração em segundos

- kMDItemTitle: nome do arquivo
- kMDItemURL: url do arquivo
- kMDItemWhereFroms: de onde foi baixado
- kMDItemUserTags: palavras chave do arquivo
- kMDItemAuthors: autor do arquivo

Na Listagem 11.5 irá buscar os arquivos que foram criados ou modificados no último dia.

#### Listagem 11.5: Exemplos de utilização do comando mdfind

```
$ mdfind cidades.txt
/Users/alessandrovivas/Documents/Livros/ProjetosIniciados/LivroMac-1aEdicao/Mac1aEdicao.pdf
/Users/alessandrovivas/Documents/Livros/ProjetosIniciados/LivroMac-1aEdicao/manipulacaoTextomac.aux
/Users/alessandrovivas/Documents/Livros/ProjetosIniciados/LivroMac-1aEdicao/sistemamac.aux
...
...
```

Procurar as mesmas condições no diretório corrente, Listagem 11.6.

#### Listagem 11.6: Exemplos de utilização do comando mdfind

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ mdfind -onlyin ./ '(
    kMDItemContentCreationDate > )
)
/Users/alessandrovivas/Documents/Livros/LivroMac-1aEdicao/desempenhomac.aux
/Users/alessandrovivas/Documents/Livros/LivroMac-1aEdicao/sistemadeArquivosmac.aux
/Users/alessandrovivas/Documents/Livros/LivroMac-1aEdicao/Mac1aEdicao.mtc13
/Users/alessandrovivas/Documents/Livros/LivroMac-1aEdicao/Mac1aEdicao.mtc14
/Users/alessandrovivas/Documents/Livros/LivroMac-1aEdicao/desempenhomac.tex
/Users/alessandrovivas/Documents/Livros/LivroMac-1aEdicao/profile
```

## 11.4 Comando units

O comando **units** é utilizado para realizar conversões entre unidades. A Listagem 11.7 mostra como converter de metros para quilômetros.

#### Listagem 11.7: Exemplos de utilização do comando units

```
$ units 5inches cm
      * 12.7
      / 0.078740157
$ units 1mile km
      * 1.609344
      / 0.62137119
$ units
Currency exchange rates from www.timegenie.com on 2014-04-02
2866 units, 109 prefixes, 79 nonlinear units

You have: 10 ounces
You want: grams
      * 283.49523
      / 0.0035273962
```

## 11.5 Comando yes

O comando **yes** é utilizado para responder automaticamente a perguntas em scripts. A Listagem 11.8 mostra um exemplo de utilização. Para terminar digite <Ctrl+C>.

### Listagem 11.8: Exemplo de utilização do comando yes para responder automaticamente a perguntas com ‘yes’

```
$ touch file1 file2 file3 && yes | rm -i file1 file2 file3
```

Ele pode ser utilizado para imprimir mensagens repetidas indefinidamente em seu terminal como na Listagem 11.9.

### Listagem 11.9: Imprimindo uma mensagem indefinidamente no terminal utilizando o yes

```
$ yes `hoje é sexta!'  
hoje é sexta!  
hoje é sexta!^C
```

Em um script para compilar texto em Latex utilize o comando **yes** para responder **r** quando ocorre erro no processamento. A Listagem 11.10 apresenta o exemplo do uso do comando.

### Listagem 11.10: Utilizando o Comando yes para Processamento de Latex

```
#!/bin/bash  
yes r | pdflatex artigo.tex  
bibtex biblio  
makeindex artigo
```

## 11.6 Linguagem awk

Awk é uma linguagem muito poderosa para processamento de textos organizados em colunas. O arquivo **cidades.txt** é organizado em duas colunas. Vamos supor que você deseja apenas imprimir a coluna 1. O processamento é feito com o comando **awk** e o resultado está na Listagem 11.11.

### Listagem 11.11: Imprimindo a Coluna 1

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ awk '{print $1}' cidades.txt  
RO  
RO  
RO  
RO  
RO  
RO  
RO  
RO  
....
```

A Listagem 11.13 apresenta a sintaxe para imprimir a coluna 2.

### Listagem 11.12: Imprimindo a Coluna 1

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ awk '{print $2}' cidades.txt  
Alta
```

```
Alto
Alto
Alvorada
Ariquemes
Buritis
Cabixi
Cacaualândia
```

....

---

A Listagem 11.13 apresenta a sintaxe para imprimir a coluna 2, coluna 3 e coluna 4.

#### Listagem 11.13: Imprimindo a Coluna 1

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ awk '{print $2,$3,$4}' cidades.txt
Alta Floresta D'Oeste
Alto Alegre dos
Alto Paraíso
Alvorada D'Oeste
Ariquemes
Buritis
Cabixi
Cacaualândia
Cacoal
Campo Novo de
Candeias do Jamari
....
```

---

Suponha que você queira contar o número de cidades de cada estado. Para isto utilize o código da Listagem 11.14.

#### Listagem 11.14: Contando o Número de Ocorrências por Cidade

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ awk '{cidades[$1]++; END { for(i in cidades
) print i, cidades[i] }' cidades.txt
MA 217
SP 645
PI 224
MG 853
BA 417
PR 399
RJ 92
MS 78
RN 167
MT 141
RO 52
ES 78
RR 15
RS 496
DF 1
GO 246
AC 22
TO 139
SC 293
AL 102
SE 75
AM 62
AP 16
PA 143
PB 223
CE 184
```

PE 185

musashi-2:LivroMac-1aEdicao alessandrovivas\$

---



# Comandos para Sistema de Arquivos

*Eu viajei por todo este país e falei com  
as pessoas mais preparadas, e posso  
assegurar que o processamento de  
dados é um engodo que não vai durar  
até o final do ano.*

O editor responsável pelos livros de  
economia na editora PrenticeHall, em  
1957.

## Sumário

12.1 Tamanho de Arquivos e Diretórios . . . . .	122
12.2 Espaço Disponível no Disco . . . . .	122
12.3 Criando Atalhos . . . . .	123
12.4 Entendendo Discos e Partições . . . . .	123
12.5 Imprimir Tabela de Partições do Linux . . . . .	123
12.6 Obtendo Informações sobre o Disco com fdisk . . . . .	124
12.7 Comando sfdisk . . . . .	124
12.8 Listando Informações sobre as Partições com lsblk . . . . .	124
12.9 Listando Informações sobre Permissões do Disco com lsblk . . . . .	124

## 12.1 Tamanho de Arquivos e Diretórios

Para calcular o tamanho de arquivos e diretórios usamos o comando **du** . Para listar o tamanho de arquivos em blocos use a Listagem 12.1.

### Listagem 12.1: Tamanho de Arquivos em Blocos

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ du -s *.pdf
10728 MacLaEdicao.pdf
20512 UmaAbordagemEstocastica.pdf
24 linux_tree_horizontal.pdf
```

Para listar o tamanho utilizando um formato legível para humanos use a Listagem 12.2.

### Listagem 12.2: Tamanho de Arquivos

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ du -sh *.pdf
5,2M MacLaEdicao.pdf
10M UmaAbordagemEstocastica.pdf
12K linux_tree_horizontal.pdf
```

## 12.2 Espaço Disponível no Disco

Para calcular o espaço disponível no disco usamos o comando **df** . Para listar o percentual de disco livre use a Listagem 12.3.

### Listagem 12.3: Espaço Disponível no Disco

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ df -h
Filesystem          Size  Used Avail Capacity  iused   ifree %iused
Mounted on
/dev/disk1           930Gi  851Gi  79Gi    92%  223219589  20670521    92%
/
devfs                328Ki  328Ki  0Bi     100%    1136      0  100%
/dev
map -hosts            0Bi   0Bi   0Bi     100%      0      0  100%
/net
map auto_home          0Bi   0Bi   0Bi     100%      0      0  100%
/home
localhost:/JOjd8EM0pgbVThhfkJHrQs  930Gi  930Gi  0Bi     100%      0      0  100%
/Volumes/MobileBackups
/dev/disk2s2           465Gi  407Gi  59Gi    88%  106594913  15417753    87%
/Volumes/ArquivosVivas
```

Para listar o tamanho de arquivos em blocos de 1 Mbyte use a Listagem 12.4.

### Listagem 12.4: Espaço Disponível no Disco em Blocos

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ df -m
Filesystem          1M-blocks  Used Available Capacity  iused   ifree %
Mounted on
/dev/disk1           952695  871705     80740    92%  223220581  20669529
92%   /
devfs                  0       0        0  100%    1136      0
100%   /dev
```

```

map -hosts                               0     0     0   100%     0     0
  100% /net
map auto_home                            0     0     0   100%     0     0
  100% /home
localhost:/JOjd8EM0pgbVThhfkJHrQs    952695 952695     0   100%     0     0
  100% /Volumes/MobileBackups
/dev/disk2s2                           476611 416386    60225    88% 106594913 15417753
  87% /Volumes/ArquivosVivas

```

---

## 12.3 Criando Atalhos

Atalhos (**aliases**) são arquivos vazios que são ponteiros para outros arquivos no disco. Para criar um link utilizamos o comando **ln**. Na Listagem 12.5 vamos criar um link chamado `LinkParaFirefox.png` que aponta para o arquivo `firefox.png`.

### Listagem 12.5: Criando um Link

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ ln -s firefox.png LinkParaFirefox.png
```

Para verificar se o link foi feito corretamente utilize o comando **ls** da Listagem 12.6. Como a primeira letra é **I** significa que é um link.

### Listagem 12.6: Listando o Link

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ ls -l LinkParaFirefox.png
lrwxr-xr-x  1 alessandrovivas  staff  11 30 Nov 16:46 LinkParaFirefox.png -> firefox.png
```

Este primeiro atalho é conhecido como **soft links**. Uma segunda maneira é fazer um atalho denominado **hard link** onde um arquivo é praticamente igual ao outro. Se você apagar um arquivo o outro não deixa de existir. Na Listagem 12.7 vamos criar um hard link chamado `hardfirefox.png` que aponta para o arquivo `firefox.png`.

### Listagem 12.7: Criando e Listando um Hard Link

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ ln firefox.png hardfirefox.png
musashi-2:LivroMac-1aEdicao alessandrovivas$ ls -l hardfirefox.png
-rw-r--r--@ 2 alessandrovivas  staff  53756 18 Dez 2013 hardfirefox.png
```

## 12.4 Entendendo Discos e Partições

Seu computador possui um ou mais discos rígidos que são utilizados para armazenar todas as informações (arquivos e programas). Cada disco pode ter uma ou mais partições e cada partição pode armazenar um sistema operacional diferente.

Os discos são montados no diretório `/dev` nos subdiretórios `sda`, `b`, etc.. Se seu computador tiver três discos ele terá os dispositivos `/dev/sda` para o disco 1, `/dev/sdb` para o disco 2 e `sdc` para o disco 3.

As partições são os valores numéricos após a letra de indicação do disco. Se seu computador tem um disco com 6 partições você terá os seguintes dispositivos `/dev/sda1`, `/dev/sda2`, `/dev/sda3`, `/dev/sda4`, `/dev/sda5` e `/dev/sda6`.

## 12.5 Imprimir Tabela de Partições do Linux

O comando **parted** pode ser utilizado para imprimir a tabela de partições de seu disco rígido. A Listagem 12.8 apresenta o procedimento para listar as partições do seu disco rígido.

#### Listagem 12.8: Listando as Partições do Disco Rígido

```
root@musashi:~# parted /dev/sda 'print'
Model: ATA WDC WD5000AAKS-7 (scsi)
Disk /dev/sda: 500GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type      File system    Flags
 1       32,3kB 197MB   197MB   primary   fat16          diag
 2       198MB   985MB   786MB   primary   ntfs
 3       985MB   262GB   261GB   primary   ntfs
 4       262GB   500GB   238GB   extended
 5       262GB   326GB   64,0GB  logical   linux-swap(v1)
 6       326GB   500GB   174GB   logical   ext4           boot
```

## 12.6 Obtendo Informações sobre o Disco com fdisk

O comando **fdisk** pode ser utilizado para obter informações detalhadas sobre suas partições. A Figura 12.1 apresenta o resultado do comando.

## 12.7 Comando sfdisk

O comando **sfdisk** pode ser utilizado para obter e realizar operações nos discos de seu computador. A Figura 12.2 apresenta o resultado do comando no disco */dev/sda*.

## 12.8 Listando Informações sobre as Partições com lsblk

O comando **lsblk** pode ser utilizado para imprimir informações sobre o seu disco rígido. A Figura 12.3 apresenta o procedimento para listar as partições do seu disco rígido.

## 12.9 Listando Informações sobre Permissões do Disco com lsblk

O comando **lsblk** pode ser utilizado para imprimir informações sobre permissões do seu disco rígido e partições. A Figura 12.4 apresenta o procedimento de utilização do comando

```
[root@musashi ~]# fdisk -l
Disco /dev/sda: 465,8 GiB, 500107862016 bytes, 976773168 setores
Unidades: setor de 1 * 512 = 512 bytes
Tamanho de setor (lógico/físico): 512 bytes / 512 bytes
Tamanho E/S (mínimo/ótimo): 512 bytes / 512 bytes
Tipo de rótulo do disco: dos
Identificador do disco: 0x50000000

Dispositivo Inicializar Início Fim Setores Tamanho Id Tipo
/dev/sda1 * 63 240974 240912 117,6M 6 FAT16
/dev/sda2 241664 20758527 20516864 9,8G 7 HPFS/NTFS/exFAT
/dev/sda3 20758528 544805064 524046537 249,9G 7 HPFS/NTFS/exFAT
/dev/sda4 544806912 976773119 431966208 206G 5 Estendida
/dev/sda5 544808960 545832959 1024000 500M 83 Linux
/dev/sda6 545835008 976773119 430938112 205,5G 8e Linux LVM
```

```
Disco /dev/mapper/fedora-swap: 3,8 GiB, 4026531840 bytes, 7864320 setores
Unidades: setor de 1 * 512 = 512 bytes
Tamanho de setor (lógico/físico): 512 bytes / 512 bytes
Tamanho E/S (mínimo/ótimo): 512 bytes / 512 bytes
```

```
Disco /dev/mapper/fedora-root: 50 GiB, 53687091200 bytes, 104857600 setores
Unidades: setor de 1 * 512 = 512 bytes
Tamanho de setor (lógico/físico): 512 bytes / 512 bytes
Tamanho E/S (mínimo/ótimo): 512 bytes / 512 bytes
```

```
Disco /dev/mapper/fedora-home: 151,7 GiB, 162919350272 bytes, 318201856 setores
Unidades: setor de 1 * 512 = 512 bytes
Tamanho de setor (lógico/físico): 512 bytes / 512 bytes
Tamanho E/S (mínimo/ótimo): 512 bytes / 512 bytes
```

```
Disco /dev/sdb: 465,8 GiB, 500107862016 bytes, 976773168 setores
Unidades: setor de 1 * 512 = 512 bytes
Tamanho de setor (lógico/físico): 512 bytes / 512 bytes
Tamanho E/S (mínimo/ótimo): 512 bytes / 512 bytes
Tipo de rótulo do disco: gpt
Identificador do disco: 03D8F472-1B09-45D2-8667-3D232A61EEE1
```

```
Dispositivo Início Fim Setores Tamanho Tipo
/dev/sdb1 40 409639 409600 200M Sistema EFI
/dev/sdb2 409640 976510983 976101344 465,5G Apple HFS/HFS+
```

Figura 12.1: Comando fdisk

```
[root@musashi ~]# sfdisk /dev/sda -l
Disco /dev/sda: 465,8 GiB, 500107862016 bytes, 976773168 setores
Unidades: setor de 1 * 512 = 512 bytes
Tamanho de setor (lógico/físico): 512 bytes / 512 bytes
Tamanho E/S (mínimo/ótimo): 512 bytes / 512 bytes
Tipo de rótulo do disco: dos
Identificador do disco: 0x50000000

Dispositivo Inicializar Início Fim Setores Tamanho Id Tipo
/dev/sda1 * 63 240974 240912 117,6M 6 FAT16
/dev/sda2 241664 20758527 20516864 9,8G 7 HPFS/NTFS/exFAT
/dev/sda3 20758528 544805064 524046537 249,9G 7 HPFS/NTFS/exFAT
/dev/sda4 544806912 976773119 431966208 206G 5 Estendida
/dev/sda5 544808960 545832959 1024000 500M 83 Linux
/dev/sda6 545835008 976773119 430938112 205,5G 8e Linux LVM
```

Figura 12.2: Comando sfdisk

```
root@musashi:~# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda 8:0 0 465,8G 0 disk
└─sda1 8:1 0 188,2M 0 part
└─sda2 8:2 0 750M 0 part
└─sda3 8:3 0 243,2G 0 part
└─sda4 8:4 0 1K 0 part
└─sda5 8:5 0 59,6G 0 part [SWAP]
└─sda6 8:6 0 162G 0 part /
sr0 11:0 1 1024M 0 rom
```

Figura 12.3: Comando lsblk

```
root@musashi:~# lsblk -m
NAME      SIZE OWNER GROUP MODE
sda      465,8G root  disk brw-rw----
└─sda1   188,2M root  disk brw-rw----
└─sda2    750M root  disk brw-rw----
└─sda3  243,2G root  disk brw-rw----
└─sda4     1K root  disk brw-rw----
└─sda5   59,6G root  disk brw-rw----
└─sda6  162G root  disk brw-rw----
sr0     1024M root cdrom brw-rw----
```

Figura 12.4: Listando Permissões com lsblk

# Comandos para Analisar o Desempenho

*Compras a distância nunca serão populares.*

Revista TIME, 1966

## Sumário

13.1 Analisando Consumo de CPU com o Comando sar . . . . .	128
13.2 Analisando de Desempenho com iostat . . . . .	128
13.3 Informações de Memória de Processos em Execução . . . . .	128
13.4 Analisando a Memória com vm_stat . . . . .	130
13.5 Comando top . . . . .	130
13.6 Comando powermetrics . . . . .	131

## 13.1 Analisando Consumo de CPU com o Comando sar

O comando **sar** é utilizado para medir o consumo de CPU de qualquer computador. O comando **sar -u** precisa de dois argumentos: a) o número de segundos entre cada leitura e b) o número de leituras que serão realizadas. A Listagem 13.1 apresenta o relatório de consumo de CPU medida a cada 1 segundo durante 10 medições.

Listagem 13.1: Analisando Desempenho da CPU com sar

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ sar -u 1 10
```

```
07:44:15 %usr %nice %sys %idle
07:44:16 9 0 6 85
07:44:17 8 0 6 86
07:44:18 9 0 6 84
07:44:19 9 0 6 85
07:44:20 9 0 7 84
07:44:21 10 0 7 83
07:44:22 8 0 6 86
07:44:23 10 0 6 84
07:44:24 9 0 6 84
07:44:25 8 0 6 86
Average: 8 0 6 84
```

Onde **%usr** é a quantidade de CPU utilizada pelo sistema com processos dos usuários, **%sys** é o percentual de processo consumido por processos do sistema, **%idle** é o percentual de CPU ocioso e **%nice** é o percentual de CPU consumidos por processos que tenham algum tipo de prioridade de escalonamento.

## 13.2 Analisando de Desempenho com iostat

O comando **iostat** é utilizado para mostrar estatísticas de Entrada e Saída e operações de dispositivos e CPU. A Listagem 13.2 apresenta o relatório com 10 medições.

Listagem 13.2: Analisando Desempenho de Todos os Núcleos com mpstat

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ iostat -c 10
              disk0          disk2        cpu    load average
             KB/t   tps  MB/s   KB/t   tps  MB/s  us sy id  1m   5m   15m
 16.88 17 0.29 121.06 1 0.07 14 7 79 1.36 2.18 2.43
 0.00 0 0.00 0.00 0 0.00 16 7 76 1.36 2.18 2.43
 0.00 0 0.00 0.00 0 0.00 15 7 78 1.25 2.15 2.41
 0.00 0 0.00 0.00 0 0.00 19 9 72 1.25 2.15 2.41
 63.80 41 2.54 0.00 0 0.00 26 9 64 1.25 2.15 2.41
 0.00 0 0.00 0.00 0 0.00 17 9 74 1.25 2.15 2.41
 0.00 0 0.00 0.00 0 0.00 18 9 74 1.25 2.15 2.41
 0.00 0 0.00 0.00 0 0.00 17 9 74 1.71 2.23 2.44
 0.00 0 0.00 0.00 0 0.00 19 9 72 1.71 2.23 2.44
 0.00 0 0.00 0.00 0 0.00 17 9 74 1.71 2.23 2.44
```

## 13.3 Informações de Memória de Processos em Execução

O comando **vmmap** mostra todas as informações sobre a memória ocupada por um determinado processo. Este comando recebe como argumento o PID do processo. Para descobrir o PID utilize o comando ps -afe. A Listagem 13.3 apresenta o procedimento para uso do comando.

### Listagem 13.3: Analisando Memória com vmmap

```
% descobrir o PID do texmaker
musashi-2:LivroMac-laEdicao alessandrovivas$ ps -afe | grep texmaker
 501  828   1   0 Qui07    ??        18:58.26 /Applications/texmaker.app/Contents/MacOS/texmaker

% rodar o comando
musashi-2:LivroMac-laEdicao alessandrovivas$ vmmap 828
2015-11-30 14:13:03.416 vmmap32[80740:1479623] *** Symbolication: Don't know how to introspect target process's malloc zone named
JavaScriptCore FastMalloc

Process:      texmaker [828]
Path:         /Applications/texmaker.app/Contents/MacOS/texmaker
Load Address: 0x1000
Identifier:   texmaker
Version:      4.3 (4.3)
Code Type:    X86
Parent Process: ??? [1]

Date/Time:    2015-11-30 14:13:02,604 -0200
Launch Time:  2015-11-26 07:09:05,133 -0200
OS Version:   Mac OS X 10.11.1 (15B42)
Report Version: 7
Analysis Tool: /Applications/Xcode.app/Contents/Developer/usr/bin/vmmap32
Analysis Tool Version: Xcode 7.1.1 (7B1005)

Virtual Memory Map of process 828 (texmaker)
Output report format: 2.4 — 32-bit process
VM page size: 4096 bytes
...
==== Summary for process 828
ReadOnly portion of Libraries: Total=279.2M resident=0K(0%) swapped_out_or_unallocated=279.2M(100%)
Writable regions: Total=236.1M written=0K(0%) resident=0K(0%) swapped_out=0K(0%) unallocated=236.1M(100%)

REGION TYPE          VIRTUAL  REGION
SIZE      COUNT (non-coalesced)
=====  =====
ATS (font support)  39.4M    10
Accelerate.framework 256K     3
Activity Tracing    2048K   2
CG backing stores   4368K   6
CG image             472K    66
CG shared images    336K    10
CoreAnimation        356K    82
CoreAnimation (empty) 608K   103
CoreUI image data   740K    71
CoreUI image file   192K    4
Foundation           40K     3
IOKit                7984K   8
Image IO             12K     4
JS garbage collector 768K    4
Kernel Alloc Once   4K     2
MALLOC guard page   48K    10
MALLOC metadata     416K   18
MALLOC_LARGE         5172K   11
MALLOC_LARGE (empty) 4240K   10
MALLOC_LARGE metadata 4K     2
MALLOC_SMALL          54.9M   84
MALLOC_SMALL (empty)  25.1M   71
MALLOC_TINY           55.0M   23
MALLOC_TINY (empty)   2048K   3
Memory Tag 242       12K     2
OpenCL               40K     6
OpenGL GLSL          128K   3
STACK GUARD          56.0M   8
Stack                10.6M   8
VM_ALLOCATE          1648K   55
VM_ALLOCATE (reserved) 12.2M   7
                           reserved VM address space (unallocated)
__DATA                12.0M   295
__GLSLBUILTINS       2588K   2
__IMAGE               528K    2
__IMPORT              104K   20
__LINKEDIT            59.2M   47
__OBJC                4548K   101
__TEXT                220.1M  300
__UNICODE              552K    2
mapped file           87.5M   59
shared memory          16.4M   12
=====
TOTAL                 687.7M  1498
TOTAL, minus reserved VM space 675.5M  1498

VIRTUAL ALLOCATION      BYTES      REGION
SIZE      COUNT  ALLOCATED % FULL  COUNT
=====  =====  =====  =====
MALLOC ZONE
=====  =====  =====  =====
DefaultMallocZone_0x3164000 102.1M  956580   46.2M  45%    181
GFXMallocZone_0x31e9000   36.0M   49      3616    0%     6
DispatchContinuations_0x31f8000 4096K  159      10K    0%     1
```

```

QuartzCore_0x4021a00          936K     919      19K    2%   178
CoreGraphics_0x4812a00         332K    8003     269K   81%    44
x-alloc_0x585e400             16K      5      160    0%     2
x-alloc_0x4869600             8K      4      224    2%     2
DefaultPurgeableMallocZone_0x505b000  0K      0      0K    0%     0
=====
TOTAL

```

---

## 13.4 Analisando a Memória com vm\_stat

Listagem 13.4: Analisando a Memória com vm\_stat

```

$ vm_stat -c 5
Mach Virtual Memory Statistics: (page size of 4096 bytes)
      free      active      specul      inactive      throttle      wired      prgble      faults      copy      0fill      reactive      purged      file-backed      anonymous
      cmprssed      cmprssor      dcmprs      comprs      pageins      pageout      swapins      swapouts
  627443  2405915  362634  283221          0  514661  115323  185938K  1309331  24597401      836      896    819123  2232647
      0      0      0          0  1162355          0      0      0      0      0      0      0      0      0
  626592  2405083  363008  283317          0  515753  115323  85428      38      3248      0      0    819500  2231908
      0      0      0          0      3      0      0      0      0      0      0      0      0      0
  626649  2406337  363009  283317          0  514342  115323  82921      71      1499      0      0    819501  2233162
      0      0      0          0      0      0      0      0      0      0      0      0      0      0
  626361  2405981  363007  283319          0  515015  115323  83764      35      1718      0      0    819501  2232806
      0      0      0          0      0      0      0      0      0      0      0      0      0      0
  625807  2407119  363008  283320          0  514342  115356  84127      70      1478      0      0    819502  2233945
      0      0      0          0      0      0      0      0      0      0      0      0      0      0

```

---

## 13.5 Comando top

O comando **top** é utilizado para fornecer estatísticas de uso de CPU, memória e diversas estatísticas de uso. A Figura 13.1 apresenta o relatório das estatísticas de uso de CPU. Para sair digite **q**.

```

top - 15:41:04 up 1 day, 1:18, 3 users, load average: 0,00, 0,01, 0,05
Tasks: 216 total, 1 running, 215 sleeping, 0 stopped, 0 zombie
%Cpu0 : 0,0 us, 0,3 sy, 0,0 ni, 99,7 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
%Cpu1 : 0,0 us, 0,0 sy, 0,0 ni, 97,0 id, 3,0 wa, 0,0 hi, 0,0 si, 0,0 st
%Cpu2 : 0,0 us, 0,0 sy, 0,0 ni, 100,0 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
%Cpu3 : 0,0 us, 0,0 sy, 0,0 ni, 99,3 id, 0,7 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem : 3843788 total, 1452360 free, 950372 used, 1441056 buff/cache
KiB Swap: 3932156 total, 3932156 free, 0 used. 2618640 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	194536	9124	4204	S	0,0	0,2	0:04.09	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.04	kthreadd
3	root	20	0	0	0	0	S	0,0	0,0	0:00.51	ksoftirqd/0
5	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	kworker/0:0H
7	root	20	0	0	0	0	S	0,0	0,0	0:53.32	rcu_sched
8	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_bh
9	root	20	0	0	0	0	S	0,0	0,0	0:17.99	rcuos/0
10	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcuob/0
11	root	rt	0	0	0	0	S	0,0	0,0	0:00.01	migration/0
12	root	rt	0	0	0	0	S	0,0	0,0	0:00.30	watchdog/0
13	root	rt	0	0	0	0	S	0,0	0,0	0:00.29	watchdog/1
14	root	rt	0	0	0	0	S	0,0	0,0	0:00.01	migration/1
15	root	20	0	0	0	0	S	0,0	0,0	0:00.51	ksoftirqd/1
17	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	kworker/1:0H
18	root	20	0	0	0	0	S	0,0	0,0	0:06.36	rcuos/1
19	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcuob/1
20	root	rt	0	0	0	0	S	0,0	0,0	0:00.27	watchdog/2
21	root	rt	0	0	0	0	S	0,0	0,0	0:00.01	migration/2
22	root	20	0	0	0	0	S	0,0	0,0	0:00.34	ksoftirqd/2
24	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	kworker/2:0H
25	root	20	0	0	0	0	S	0,0	0,0	0:15.34	rcuos/2
26	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcuob/2
27	root	rt	0	0	0	0	S	0,0	0,0	0:00.25	watchdog/3
28	root	rt	0	0	0	0	S	0,0	0,0	0:00.01	migration/3
29	root	20	0	0	0	0	S	0,0	0,0	0:00.33	ksoftirqd/3

Figura 13.1: Uso do top para Obter Estatísticas de CPU

## 13.6 Comando powermetrics

O comando **powermetrics** [18] mostra estatísticas do uso de energia em seu computador. A Listagem 13.5 apresenta a sintaxe do comando.

### Listagem 13.5: Analisando Informações sobre Energia

```
musashi-2:LivroMac-laEdicao alessandrovivas$ sudo powermetrics -a --show-usage-summary
Password:
unable to get smc values
Machine model: MacBookPro8,1
SMC version: 1.68 f99
EFI version: MBP81.0047.B2A
OS version: 15B42
Boot arguments:
Boot time: Thu Nov 26 07:00:16 2015

*** Sampled system activity (Tue Dec 1 07:39:03 2015 -0200) (5008.97ms elapsed) ***

*** Running tasks ***

Name           ID    CPU ms/s  User%  Deadlines (<2 ms, 2-5 ms)  Wakeups (Intr, Pkg idle)
VTDecoderXPCService  74357  374.90   95.50    0.00      0.00          0.00      0.00

**** Battery and backlight usage ****

Backlight level: 978 (range 0-1024)
Keyboard Backlight level: 0

**** Network activity ****

out: 56.10 packets/s, 5741.69 bytes/s
in: 66.88 packets/s, 80378.93 bytes/s

**** Disk activity ****

read: 2.00 ops/s 14.72 KBytes/s
write: 65.48 ops/s 1168.13 KBytes/s

**** Interrupt distribution ****

CPU 0:
Vector 0x57(EHC2): 1030.35 interrupts/sec
Vector 0x70(IGPU): 302.26 interrupts/sec
Vector 0x72(HDEF): 3.59 interrupts/sec
Vector 0x73(SATA): 67.28 interrupts/sec
Vector 0x74(APPT): 172.29 interrupts/sec
Vector 0xdd(TMR): 731.89 interrupts/sec
Vector 0xde(IPI): 310.04 interrupts/sec
CPU 1:
Vector 0xdd(TMR): 185.07 interrupts/sec
Vector 0xde(IPI): 324.62 interrupts/sec
CPU 2:
Vector 0xdd(TMR): 294.67 interrupts/sec
Vector 0xde(IPI): 545.22 interrupts/sec
CPU 3:
Vector 0xdd(TMR): 129.37 interrupts/sec
Vector 0xde(IPI): 364.35 interrupts/sec

**** Processor usage ****

Intel energy model derived package power (CPUs+GT+SA): 14.36W
```



# Configuração de Hardware e Software

*640 KB serão suficientes para qualquer um no futuro.*

Bill Gates , 1981

## Sumário

14.1 Visualizando Informações de Boot . . . . .	134
14.2 Visualizando Informações sobre o Sistema . . . . .	134
14.3 Visualizando Informações sobre a Versão do Kernel . . . . .	134
14.4 Informações sobre a Versão do Software . . . . .	135
14.5 Informações de Hardware com hostinfo . . . . .	135
14.6 Informações sobre o Sistema . . . . .	135
14.7 Visualizando Informações sobre a sua CPU . . . . .	136
14.8 Visualizando todas as Variáveis do Kernel . . . . .	137
14.9 Visualizando Informações sobre os Dispositivos USB . . . . .	138
14.10 Listando Todos os Dispositivos PCI . . . . .	140
14.11 Verificando Todas as Partições . . . . .	140
14.12 Obtendo Informações sobre a Memória . . . . .	141
14.13 Listando Impressoras do Sistema . . . . .	141

## 14.1 Visualizando Informações de Boot

O comando **bless** pode ser utilizado para mostrar informações sobre o sistema de boot no **Mac OS X** como na Listagem 14.1.

### Listagem 14.1: Informações sobre boot

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ bless -info
finderinfo[0]: 24679865 => Blessed System Folder is /System/Library/CoreServices
3finderinfo[1]: 27814931 => Blessed System File is /System/Library/CoreServices/boot.efi
finderinfo[2]:      0 => Open-folder linked list empty
finderinfo[3]:      0 => No alternate OS blessed file/folder
6finderinfo[4]:      0 => Unused field unset
finderinfo[5]: 24679865 => OS X blessed folder is /System/Library/CoreServices
64-bit VSDB volume id: 0x09C88D01C36256C3
```

Para descobrir o disco de boot utilize a Listagem 14.2 [2].

### Listagem 14.2: Disco de boot

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ bless -info -getBoot
/dev/disk1
```

## 14.2 Visualizando Informações sobre o Sistema

O comando **dmesg** apresenta várias mensagens de log de sistema do **Mac OS X**. Ele deve ser executado como administrador e mostra as diversas mensagens de erro do sistema. A Listagem 14.3 apresenta o resultado do comando.

### Listagem 14.3: Resultado do Comando dmesg

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ sudo dmesg
Password:
3, this increases system memory footprint until the target exits.
Google Chrome He[71601] triggered unnest of range 0x7fff85000000->0x7fff85200000 of DYLD
    shared region in VM map 0x7a904bf6c5af1e51. While not abnormal for debuggers, this
    increases system memory footprint until the target exits.
SerialATAPI device reconfiguration did not complete successfully. (failedCommandInfo = 0
    x1)
6SerialATAPI device reconfiguration did not complete successfully. (failedCommandInfo = 0
    x1)
Google Chrome He[71850] triggered unnest of range 0x7fff85000000->0x7fff85200000 of DYLD
    shared region in VM map 0x7a904bf6aed8f9f1. While not abnormal for debuggers, this
    increases system memory footprint until the target exits.
Google Chrome He[71854] triggered unnest of range 0x7fff85000000->0x7fff85200000 of DYLD
    shared region in VM map 0x7a904bf6aed8f709. While not abnormal for debuggers, this
    increases system memory footprint until the target exits.
9...
```

## 14.3 Visualizando Informações sobre a Versão do Kernel

Para verificar informações sobre o seu computador utilize o comando **uname** como na Listagem 14.4.

#### Listagem 14.4: Versão do Kernel

```
[musashi-2:LivroMac-1aEdicao alessandrovivas$ uname -a
Darwin musashi-2.local 15.0.0 Darwin Kernel Version 15.0.0: Sat Sep 19 15:53:46 PDT 2015;
root:xnu-3247.10.11~1/RELEASE_X86_64 x86_64
```

Para verificar se sua arquitetura é 32 ou 64 bits utilize o comando da Listagem 14.5.

#### Listagem 14.5: Arquitetura do Hardware

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ uname -m
x86_64
```

## 14.4 Informações sobre a Versão do Software

O comando **sw\_vers** imprime as informações sobre a versão do **Mac OS X**. A sintaxe está explicada na Listagem 14.6.

#### Listagem 14.6: Informações do Software

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ sw_vers
ProductName: Mac OS X
ProductVersion: 10.11.1
BuildVersion: 15B42
```

## 14.5 Informações de Hardware com hostinfo

O comando **hostinfo** imprime as informações gerais sobre o seu sistema. A sintaxe está explicada na Listagem 14.7.

#### Listagem 14.7: Informações Gerais sobre seu Sistema

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ hostinfo
Mach kernel version:
3 Darwin Kernel Version 15.0.0: Sat Sep 19 15:53:46 PDT 2015; root:xnu-3247.10.11~1/
RELEASE_X86_64
Kernel configured for up to 4 processors.
2 processors are physically available.
64 processors are logically available.
Processor type: i486 (Intel 80486)
Processors active: 0 1 2 3
9Primary memory available: 16.00 gigabytes
Default processor set: 321 tasks, 3033 threads, 4 processors
Load average: 1.76, Mach factor: 2.32
```

## 14.6 Informações sobre o Sistema

O comando **system\_profiler** lista informações simples sobre as configurações de seu sistema. A Listagem 14.8 apresenta o resultado do comando em minha máquina. O arquivo gerado é bem completo (3.4 Mb no meu teste ) e

possui todas as informações.

#### Listagem 14.8: Informações do Sistema

```
Accessibility:  
3     Accessibility Information:  
      Cursor Magnification: Off  
6     Display: Black on White  
     Flash Screen: Off  
    Mouse Keys: Off  
9     Slow Keys: Off  
   Sticky Keys: Off  
VoiceOver: Off  
12    Zoom Mode: Full Screen  
Contrast: 0  
Keyboard Zoom: On  
15    Scroll Zoom: Off  
....
```

## 14.7 Visualizando Informações sobre a sua CPU

O comando **sysctl** é utilizado para listar todas as informações sobre a sua CPU. A Figura 14.9 apresenta o resultado do comando.

#### Listagem 14.9: Informações do CPU

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ sysctl hw  
hw.ncpu: 4  
3hw.byteorder: 1234  
hw.memsize: 17179869184  
hw.activecpu: 4  
6hw.targettype:  
hw.physicalcpu: 2  
hw.physicalcpu_max: 2  
9hw.logicalcpu: 4  
hw.logicalcpu_max: 4  
hw.cputype: 7  
12hw.cpusubtype: 4  
hw.cpu64bit_capable: 1  
hw.cpufamily: 1418770316  
15hw.cacheconfig: 4 2 2 4 0 0 0 0 0 0  
hw.cachesize: 17179869184 32768 262144 3145728 0 0 0 0 0 0  
hw.pagesize: 4096  
18hw.pagesize32: 4096  
hw.busfrequency: 1000000000  
hw.busfrequency_min: 1000000000  
21hw.busfrequency_max: 1000000000  
hw.cpufrequency: 2400000000  
hw.cpufrequency_min: 2400000000  
24hw.cpufrequency_max: 2400000000  
hw.cachelinesize: 64  
hw.l1icachesize: 32768  
27hw.l1dcachesize: 32768  
hw.l2cachesize: 262144  
hw.l3cachesize: 3145728  
30hw.tbfrequency: 1000000000
```

```
hw.packages: 1
hw.optional.floatingpoint: 1
33hw.optional.mmx: 1
hw.optional.sse: 1
hw.optional.sse2: 1
36hw.optional.sse3: 1
hw.optional.supplementalsse3: 1
hw.optional.sse4_1: 1
39hw.optional.sse4_2: 1
hw.optional.x86_64: 1
hw.optional.aes: 1
42hw.optional.avx1_0: 1
hw.optional.rdrand: 0
hw.optional.f16c: 0
45hw.optional.enfstrg: 0
hw.optional.fma: 0
hw.optional.avx2_0: 0
48hw.optional.bmil: 0
hw.optional.bmi2: 0
hw.optional.rtm: 0
51hw.optional.hle: 0
hw.optional.adx: 1
hw.optional.mpx: 0
54hw.optional.sgx: 0
hw.cputhreadtype: 1
```

---

## 14.8 Visualizando todas as Variáveis do Kernel

O comando **sysctl** é utilizado para listar todas as variáveis do Kernel. A Figura 14.10 apresenta o resultado do comando [9].

**Listagem 14.10: Informações das Variáveis do Kernel**

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ sysctl -a
user.cs_path: /usr/bin:/bin:/usr/sbin:/sbin
3...
kern.ostype: Darwin
..
6vm.loadavg: { 2,14 7,15 11,54 }
..
vfs.generic.maxtypenum: 23
9...
net.local.stream.sendspace: 8192
...
12debug.lowpri_throttle_max_iosize: 131072
...
hw.ncpu: 4
15...
```

---

Caso queira uma variável específica basta passar como argumento o nome da variável como na Listagem 14.11.

**Listagem 14.11: Imprimindo uma Variável Específica**

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ sysctl net
net.local.stream.sendspace: 8192
net.local.stream.recvspace: 8192
net.local.stream.tracemdns: 0
```

```
net.local.dgram.maxdgram: 2048
net.local.dgram.recvspace: 4096
...
```

---

## 14.9 Visualizando Informações sobre os Dispositivos USB

O comando **ioreg** é utilizado para listar várias informações e pode ser utilizado para visualizar as informações sobre as conexões USB [3]. A Listagem 14.12 apresenta o resultado do comando.

### Listagem 14.12: Lista Dispositivos USB

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ ioreg -p IOUSB -w0 | sed 's/[^\o]*\o //; s/@.*$//' | grep -v '^Root.*'
IOUSBHostDevice
IR Receiver
FaceTime HD Camera (Built-in)
IOUSBHostDevice
BRCM2070 Hub
Bluetooth USB Host Controller
Apple Internal Keyboard / Trackpad
```

Outro comando que pode ser utilizado para o mesmo fim é o **system\_profiler** [17]. A Listagem 14.13 apresenta a sintaxe e o resultado.

### Listagem 14.13: Listando USB

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ system_profiler SPUSBDataType
USB:

USB 2.0 Bus:

    Host Controller Driver: AppleUSBEHCIPCI
    PCI Device ID: 0x1c2d
    PCI Revision ID: 0x0005
    PCI Vendor ID: 0x8086

    FaceTime HD Camera (Built-in):

        Product ID: 0x8509
        Vendor ID: 0x05ac (Apple Inc.)
        Version: 5.16
        Serial Number: CC2C4A012BDG6LL0
        Speed: Up to 480 Mb/sec
        Manufacturer: Apple Inc.
        Location ID: 0xfa200000 / 1
        Current Available (mA): 1000
        Current Required (mA): 500
        Extra Operating Current (mA): 0
        Built-In: Yes

    Hub:

        Product ID: 0x2513
        Vendor ID: 0x0424 (SMSC)
        Version: b.b3
        Speed: Up to 480 Mb/sec
```

Location ID: 0xfa100000 / 2  
Current Available (mA): 1000  
Current Required (mA): 2  
Extra Operating Current (mA): 0  
Built-In: Yes

Apple Internal Keyboard / Trackpad:

Product ID: 0x0252  
Vendor ID: 0x05ac (Apple Inc.)  
Version: 1.18  
Speed: Up to 12 Mb/sec  
Manufacturer: Apple Inc.  
Location ID: 0xfa120000 / 5  
Current Available (mA): 1000  
Current Required (mA): 40  
Extra Operating Current (mA): 0  
Built-In: Yes

BRCM2070 Hub:

Product ID: 0x4500  
Vendor ID: 0x0a5c (Broadcom Corp.)  
Version: 1.00  
Speed: Up to 12 Mb/sec  
Manufacturer: Apple Inc.  
Location ID: 0xfa110000 / 4  
Current Available (mA): 1000  
Current Required (mA): 94  
Extra Operating Current (mA): 0  
Built-In: Yes

Bluetooth USB Host Controller:

Product ID: 0x821a  
Vendor ID: 0x05ac (Apple Inc.)  
Version: 0.42  
Speed: Up to 12 Mb/sec  
Manufacturer: Apple Inc.  
Location ID: 0xfa113000 / 8  
Current Available (mA): 1000  
Current Required (mA): 0  
Extra Operating Current (mA): 0  
Built-In: Yes

USB 2.0 Bus:

Host Controller Driver: AppleUSBEHCIPCI  
PCI Device ID: 0x1c26  
PCI Revision ID: 0x0005  
PCI Vendor ID: 0x8086

Hub:

Product ID: 0x2513  
Vendor ID: 0x0424 (SMSC)  
Version: b.b3  
Speed: Up to 480 Mb/sec  
Location ID: 0xfd100000 / 1  
Current Available (mA): 1000  
Current Required (mA): 2

```
Extra Operating Current (mA): 0
Built-In: Yes

IR Receiver:

Product ID: 0x8242
Vendor ID: 0x05ac (Apple Inc.)
Version: 0.16
Speed: Up to 1.5 Mb/sec
Manufacturer: Apple Computer, Inc.
Location ID: 0xfd110000 / 2
Current Available (mA): 1000
Current Required (mA): 100
Extra Operating Current (mA): 0
Built-In: Yes
```

---

## 14.10 Listando Todos os Dispositivos PCI

O comando **ioreg** pode ser utilizado para listar todas as informações sobre os dispositivos PCI. A Listagem 14.14 apresenta o resultado do comando.

### Listagem 14.14: Listando PCI

```
$ ioreg -l | grep PCI
|   IOKitDiagnostics = {Container allocation=9214076, Instance allocation=9818272,
|     Pageable allocation=1380359504, Classes={IONDRVFramebuffer=0, AppleSNBFBUserClient=1,
|       IOKitDiagnosticsClient=0, IONaturalMemoryCursor=1,
|       IOAudioClientBufferSet=2, DspFuncBuzzKill=0, AppleFWOHCI_PM_PPC=0,
|       AppleHDAMikeyInternalCS4208=0, IOFireWireDevice=0,
|       AppleHDAUDIO_Codec=0, IOGen575Shared=1, IOHDACodecDevice=2, AppleAHCIWorkLoop=2,
|       IOThunderboltDeficitCommandQueue=1,
|       CoreStorageLogical=1, IORRegistryEntry=33, AppleAHCIPort=1, AppleHDAWorkLoop=1,
|       SMCMotionSensor=1, IOHIDevice=2,
|       IOFWUserWriteCommand=0, IOPCIEEventSource=0, IOThunderboltAbstractMicro=1, DspFun
...
...
```

O comando **system\_profiler** também pode ser utilizado para o mesmo fim como visto na Listagem 14.15.

### Listagem 14.15: Listando PCI

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ system_profiler | grep PCI
Bus: PCI
AppleSMBusPCI:
  Bundle ID: com.apple.driver.AppleSMBusPCI
  Get Info String: AppleSMBusPCI version 1.0.14, Copyright 2008-2015 Apple Inc.
  Location: /System/Library/Extensions/AppleSMBusPCI.kext
AppleThunderboltPCIAapters:
  Bundle ID: com.apple.driver.AppleThunderboltPCIAapters
  Get Info String: AppleThunderboltPCIAapters version 2.0.2, Copyright Â© 2009-2014
    Apple Inc. All rights reserved.
...
```

## 14.11 Verificando Todas as Partições

Com o comando **diskutil** [1] é possível visualizar as informações sobre as partições em seu disco. A Listagem 14.16 apresenta o resultado do comando.

#### Listagem 14.16: Imprimindo as Partições

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ diskutil list
/dev/disk0 (internal, physical):
 #:          TYPE NAME               SIZE      IDENTIFIER
 0: GUID_partition_scheme           *1.0 TB   disk0
    1:   EFI   EFI                  209.7 MB  disk0s1
    2: Apple_CoreStorage Macintosh HD 999.3 GB  disk0s2
    3:   Apple_Boot Recovery HD     650.0 MB  disk0s3
/dev/disk1 (internal, virtual):
 #:          TYPE NAME               SIZE      IDENTIFIER
 0: Apple_HFS Macintosh HD        +999.0 GB  disk1
   Logical Volume on disk0s2
   2606D3D7-DF04-493B-8EE5-14B31272FBFD
   Unlocked Encrypted
```

## 14.12 Obtendo Informações sobre a Memória

O comando **hostinfo** lista a quantidade de memória disponível em seu computador. A Listagem 14.17 apresenta o resultado do comando.

#### Listagem 14.17: Listando Informações sobre a Memória

```
[musashi-2:LivroMac-1aEdicao alessandrovivas$ hostinfo | grep memory
Primary memory available: 16.00 gigabytes
```

## 14.13 Listando Impressoras do Sistema

O comando **lpstat** lista as impressoras que estão instaladas em seu computador. A Listagem 14.18 apresenta o resultado do comando.

#### Listagem 14.18: Listando Informações sobre Impressoras

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ lpstat -a
Brother_HL_5370DW_series aceitando solicitações desde Qua 11 Nov 15:16:32 2015
Brother_HL_5370DW_series_2 aceitando solicitações desde Qui  7 Mar 15:04:00 2013
HP_LaserJet_P2035 aceitando solicitações desde Seg 30 Nov 22:49:47 2015
hp_psc_1300_series aceitando solicitações desde Qui 12 Nov 19:05:32 2015
hp_psc_1300_series_2 aceitando solicitações desde Qui 22 Jan 08:26:30 2015
```

Para listar a impressora padrão utilize a Listagem 14.19.

#### Listagem 14.19: Listando a Impressora Padrão

```
musashi-2:LivroMac-1aEdicao alessandrovivas$ lpstat -d
destino padrão de sistema: HP_LaserJet_P2035
```

# Índice

## Análise de Desempenho

- iostat, 128
- powermetrics, 131
- sar, 128
- top, 130
- vmmmap, 129

## Comandos Úteis

- awk, 117
- mdfind, 116
- mtls, 114
- unit, 116
- yes, 117

## Comandos de Manipulação de Arquivos e Diretórios

- cd, 19, 20
- cp, 21
- echo, 6
- mkdir, 24
- mv, 22
- pwd, 20
- rm, 23
- rmdir, 23
- touch, 23

## Comandos de Processamento de Texto

- cat, 4, 26
- cmp, 27
- comm, 27
- csplit, 28
- cut, 29
- echo, 4
- expand, 30
- fmt, 32
- grep, 33
- head, 33
- iconv, 33
- look, 34
- more, 35
- nl, 35
- paste, 29
- pr, 36
- rev, 36
- sort, 37

- stat, 37

- tail, 37
- unexpand, 31
- uniq, 38
- wc, 39

## Comandos de Rede

- hostname, 8

## Comandos de Redes

- arp, 74
- curl, 74
- dig, 80
- host, 79
- hostname, 74
- ifconfig, 75, 77, 78
- netstat, 82
- nettop, 93
- newtorksetup, 75
- nmap, 93
- nslookup, 81
- ping, 79
- route, 96
- scp, 101
- ssh, 101
- tcpdump, 102
- telnet, 97
- traceroute, 81
- wget, 105

## Comandos de Sistema

- alias, 42
- basename, 42
- builtin, 42
- cal, 47
- calendar, 49
- chflags, 49
- chsh, 5
- clear, 3, 6
- compgen, 43
- complete, 43
- cron, 43
- crontab, 43
- date, 47
- dscl, 44

exit, 12  
find, 54  
finger, 49  
halt, 14  
history, 9  
HISTSIZE, 10  
id, 46  
last, 50  
locate, 54  
logout, 12  
ls, 5, 16  
passwd, 47  
PATH, 52  
pipe, 33  
pwd, 44  
reboot, 14  
shutdown, 12  
su, 51  
type, 42  
uname, 51  
uptime, 52  
users, 47  
vm\_stat, 50  
w, 53  
whatis, 54  
whereis, 53  
whic, 53  
whoami, 42, 44

Comandos de Sistemas de Arquivos  
df, 122  
du, 122  
fdisk, 124  
ln, 123  
lsblk, 124  
parted, 124  
sfdisk, 124

Comandos Diversos  
say, 114

Compactação e Backup  
asr, 111  
bzip2, 110  
hdutil, 110  
tar, 109  
unzip, 108  
zip, 108  
zipinfo, 108

Controle de Processos  
kill, 13

Gerenciamento de Processos

bg, 58  
fg, 58  
jobs, 58  
kill, 62  
killall, 63  
ps, 59  
time, 63  
top, 61

Gerenciamento de Usuários e Grupos  
dscacheutil, 70  
dscl, 70–72  
passwd, 71  
useradd, 70

Hardware e Software  
bless, 134  
diskutil, 141  
dmesg, 134  
hostinfo, 135, 141  
ioreg, 140  
lpstat, 141  
lsusb, 138  
sw\_vers, 135  
sysctl, 136, 137  
system\_profiler, 135  
uname, 134

Modo Privilegiado  
sudo, 4, 8

Permissão e Propriedade  
chgrp, 66  
chmod, 67  
chow, 66  
chown, 67  
chroot, 68

Shells  
bash, 4  
csh, 4  
ksh, 4  
sh, 4  
Shells, 2  
tcsh, 4  
zsh, 4



# Bibliografia

- [1] osx - List all devices connected, lsblk for Mac OS X - Ask Different.
- [2] Allens. Bless System in Mac OS X, if your partition won't boot: | Allen's Windows/Linux blog on Word-Press.com.
- [3] AskDifferent. List USB devices on OSX command line - Ask Different, 2015.
- [4] Stack Exchange. mac osx - How do I create user accounts from the Terminal in Mac OS X 10.5? - Server Fault, 2015.
- [5] Server Fault. mac osx - How do I add a group in Mac OS X 10.6? - Server Fault, 2015.
- [6] GeeNeil. Compress files & folder in Terminal Command Line on OSX Yosemite, 2015.
- [7] Glazenbakje. Add or delete static routes Apple Mac OS X Mountain Lion | Glazenbakje's Weblog on Word-Press.com, 2015.
- [8] Krypted. Create Groups Using dscl | krypted.com, 2015.
- [9] Command Line. Command Line Mac: Sysctl.
- [10] Daniel Miessler. A tcpdump Primer with Examples, 2015.
- [11] NMAP. NMAP Apple Mac OS X, 2015.
- [12] OSDAILY. Watch Network Traffic in Mac OS X via Command Line with nettop, 2015.
- [13] OSXDAILY. Enable Remote Login to Start SSH Server in Mac OS X.
- [14] OSXDAILY. List All Network Hardware from the Command Line in OS X, 2015.
- [15] Regaz. How-to Delete Users and Groups from Terminal on Mac OS X, 2015.
- [16] The Geek Stuff. 15 Practical Linux cURL Command Examples (cURL Download Examples).
- [17] Apple Support. What is the terminal command to list USB devices? | Apple Support Communities.
- [18] Dave Taylor. *Learning Unix for OS X Mountain Lion*. O'Reilly Media, 2012.
- [19] Tekrevue. How to Make Your Mac Talk Using Terminal's Say Command, 2015.