

# LINUX

## Utilitários de Redes



Eduardo Maroñas Monks

[YOUTUBE.COM/EMMONKS](https://www.youtube.com/emmonks)

Este livro tem como objetivo disponibilizar um guia de referência rápido em utilitários de rede para auxiliar administradores de sistemas operacionais Linux nas rotinas de administração.

*1ª edição (revisão 1), Junho de 2023*

# Sumário

<b>1</b>	<b>Introdução .....</b>	<b>5</b>
1.1	Administração de Sistemas Linux	5
1.2	Ambiente para Testes	5
1.3	Escolha da Distribuição	5
<b>2</b>	<b>Comandos Básicos .....</b>	<b>7</b>
2.1	ifconfig	7
2.2	ip	8
2.3	arp	10
2.4	netstat	10
2.5	route	11
2.6	ethtool	12
2.7	dig	12
2.8	host	13
2.9	nslookup	14
2.10	ping	15
2.11	traceroute	15
2.12	iperf	15
2.13	arping	16
2.14	wget	17
2.15	ssh e scp	17
2.16	ss	18
2.17	mtr	18
2.18	dhclient	19
2.19	telnet	19
2.20	netcat (nc)	19
2.21	curl	20
2.22	rsync	20
2.23	iwconfig	21
2.24	iwlist	21
2.25	ntpdate	21

---

<b>3</b>	<b>Procedimentos .....</b>	<b>23</b>
<b>3.1</b>	<b>Renomear interface</b>	<b>23</b>
<b>3.2</b>	<b>Uso de VLANs</b>	<b>23</b>
<b>3.3</b>	<b>Interface em Bridge</b>	<b>24</b>
<b>3.4</b>	<b>Interface em Bonding</b>	<b>26</b>
<b>3.5</b>	<b>Configurações de interface em arquivos</b>	<b>28</b>
<b>3.6</b>	<b>iftop</b>	<b>31</b>

# 1. Introdução

Neste capítulo são apresentadas as motivações para o conhecimento das ferramentas de administração de redes em sistemas operacionais Linux.

## 1.1 Administração de Sistemas Linux

A administração de servidores Linux é a rotina da maioria dos administradores de redes e sistemas. Desta forma, dominar os comandos e utilitários essenciais do Linux agilizam o diagnóstico e a resolução dos problemas. Portanto, o primeiro momento é ter conhecimento de quais são os comandos e utilitários essenciais. Em um segundo momento é explorá-los para adaptá-los as necessidades de cada problema.

## 1.2 Ambiente para Testes

Para realizar testes com as ferramentas é importante o uso com cenários mais completos. Uma das alternativas é o uso do emulador CORE. Neste emulador é possível criar cenários complexos com centenas de hosts Linux e com o uso das ferramentas apresentadas neste livro. Uma máquina virtual com o emulador CORE está disponível ([link da VM](#)).

## 1.3 Escolha da Distribuição

As principais distribuições são baseadas em Redhat (CentOS, Oracle Linux) ou em Debian (Ubuntu, Mint). As diferenças principais entre as distribuições estão na forma de gerenciamento de pacotes, na organização do sistema de arquivos com caminhos diferentes e na política de uso. Por exemplo, uma distribuição tal como a Debian possui políticas restritas para o uso de pacotes que não sigam o licenciamento GPLv3 e os não disponibiliza por padrão. Entretanto, a escolha da distribuição a ser usada é regida muito mais por uma questão de gosto pessoal ou por necessidade de homologação devido a alguma aplicação a ser instalada ou algum hardware específico.

Os utilitários tratados neste livro estão disponíveis em qualquer distribuição atual e os parâmetros dos exemplos deverão funcionar sem problemas.



## 2. Comandos Básicos

Neste capítulo são apresentados os comandos básicos para configuração de interfaces de rede. Uma das exceções aos comandos básicos é o comando **ip** que engloba diversas funcionalidades para administração de redes em Linux.

### 2.1 ifconfig

**Descrição 2.1** Comando para realizar configurações na interface de rede e obter estatísticas diversas. Em distribuições atuais vem sendo substituído pelo comando **ip**.

- Para mostrar todas as interfaces disponíveis
  - **Exemplo 2.1** `ifconfig -a`
- Para mostrar informações sobre a interface `eth0`
  - **Exemplo 2.2** `ifconfig eth0`
- Para configurar um IP na interface `eth0`
  - **Exemplo 2.3** `ifconfig eth0 10.0.0.2 netmask 255.255.255.0`
  - **Exemplo 2.4** `ifconfig eth0 10.0.0.2/24`
- Trocar o endereço físico da interface `eth0`
  - **Exemplo 2.5** `ifconfig eth0 hw ether 00:cc:00:ff:ff:ee`
- Para criar um outro IP na interface `eth0` (IP alias)
  - **Exemplo 2.6** `ifconfig eth0:1 10.10.0.2/24`
- Para modificar o MTU da interface `eth0`
  - **Exemplo 2.7** `ifconfig eth0 mtu 9000`
- Para remover um IP da interface `eth0`
  - **Exemplo 2.8** `ifconfig eth0 0.0.0.0`
- Para desativar a interface `eth0`
  - **Exemplo 2.9** `ifconfig eth0 down`
- Para desativar um IP alias
  - **Exemplo 2.10** `ifconfig eth0:1 down`
- Para ativar a interface `eth0`
  - **Exemplo 2.11** `ifconfig eth0 up`
- Para adicionar um endereço IPv6 na interface `eth0`
  - **Exemplo 2.12** `ifconfig eth0 inet6 add 2001:0db8:0:200::3/64`
- Para remover um endereço IPv6 na interface `eth0`

■ **Exemplo 2.13** `ifconfig eth0 inet6 del 2001:0db8:0:200::3/64`

## 2.2 ip

**Descrição 2.2** Comando para realizar configurações na interface de rede, roteamento, tunelamento, obter estatísticas e outras diversas funcionalidades. Em distribuições atuais está disponibilizado como principal ferramenta de configuração de interfaces de rede.

- Para listar todas as interfaces
  - **Exemplo 2.14** `ip link show`
- Para ativar a interface `eth0`
  - **Exemplo 2.15** `ip link set eth0 up`
- Para desativar a interface `eth0`
  - **Exemplo 2.16** `ip link set eth0 down`
- Para mostrar o endereçamento das interfaces
  - **Exemplo 2.17** `ip addr show`
- Para mostrar o endereçamento da interface `eth0`
  - **Exemplo 2.18** `ip addr show dev eth0`
- Para mostrar os hosts vizinhos (conectados na mesma rede física que tenham de comunicado como host, tabela ARP)
  - **Exemplo 2.19** `ip neigh show`
  - **Exemplo 2.20** `ip neigh show dev eth0`
- Para adicionar uma entrada na tabela ARP
  - **Exemplo 2.21** `ip neigh add 192.168.1.1 lladdr 00:cc:00:ff:ff:ee dev eth0`
- Para remover uma entrada da tabela ARP
  - **Exemplo 2.22** `ip neigh del 192.168.1.1 dev eth0`
- Limpar toda a tabela ARP de uma VLAN
  - **Exemplo 2.23** `ip -s neigh flush dev eth1.212`
- Para adicionar mais um IP na interface `eth0` (similar ao IP alias do `ifconfig`)
  - **Exemplo 2.24** `ip addr add 192.168.1.2/24 dev eth0`
- Para remover endereços adicionais no `eth0`
  - **Exemplo 2.25** `ip addr del 192.168.1.5/24 dev eth0`
- Para configurar um endereço IP na interface `eth0`
  - **Exemplo 2.26** `ip addr add 1.2.3.4/24 broadcast 1.2.3.255 dev eth0`
- Para remover um endereço IP na interface `eth0`
  - **Exemplo 2.27** `ip addr del 1.2.3.4/24 broadcast 1.2.3.255 dev eth0`
- Para trocar o endereço físico da interface `eth0`



- **Exemplo 2.28** ip link set dev eth0 down
- **Exemplo 2.29** ip link set dev eth0 address 00:cc:00:ff:ff:ee
- **Exemplo 2.30** ip link set dev eth0 up
- Para modificar o MTU da interface para o valor 1476
  - **Exemplo 2.31** ip link set mtu 1476 dev eth0
- Para listar a tabela de roteamento
  - **Exemplo 2.32** ip route show
- Para adicionar uma rota padrão IPv4
  - **Exemplo 2.33** ip route add default via 192.168.1.254
- Para remover uma rota padrão IPv4
  - **Exemplo 2.34** ip route del default via 192.168.1.254
- Para adicionar uma rota estática IPv4
  - **Exemplo 2.35** ip route add 192.168.1.0/24 dev eth0
  - **Exemplo 2.36** ip route add 192.168.8.0/24 via 192.168.254.254
- Para remover uma rota estática IPv4
  - **Exemplo 2.37** ip route del 192.168.1.0/24 dev eth0
  - **Exemplo 2.38** ip route del 192.168.8.0/24 via 192.168.254.254
- Para mostrar os endereços IPv6 de todas as interfaces
  - **Exemplo 2.39** ip -6 a
- Para adicionar um endereço IPv6 na interface eth0
  - **Exemplo 2.40** ip -6 addr add 2001:0db8:0:200::3/64 dev eth0
- Para remover um endereço IPv6 na interface eth0
  - **Exemplo 2.41** ip -6 addr del 2001:0db8:0:200::3/64 dev eth0
- Para adicionar uma rota default IPv6
  - **Exemplo 2.42** ip -6 route add default via 2001:0db8:0:200::1
- Para remover uma rota default IPv6
  - **Exemplo 2.43** ip -6 route del default via 2001:0db8:0:200::1
- Para adicionar uma rota estática IPv6
  - **Exemplo 2.44** ip -6 route add 2001:0db8:0:201::/64 via 2001:0db8:0:200::1
- Para remover uma rota estática IPv6
  - **Exemplo 2.45** ip -6 route del 2001:0db8:0:201::/64 via 2001:0db8:0:200::1
- Para listar a tabela de roteamento IPv6
  - **Exemplo 2.46** ip -6 route show

- Para listar a tabela de vizinhos
  - **Exemplo 2.47** `ip -6 neigh show`
- Para adicionar as marcações de VLANs. No exemplo é adicionada a VLAN ID 100 na interface eth1, criando a interface eth1.100. Para criar múltiplas interfaces marcadas basta repetir o comando variando o nome (nome) e o vlan id.
  - **Exemplo 2.48** `ip link add link eth1 name eth1.100 type vlan id 100`
- Para remover uma interface marcada para uso de VLANs. No exemplo
  - **Exemplo 2.49** `ip link delete eth1.100`

## 2.3 arp

**Descrição 2.3** O protocolo ARP (*Address Resolution Protocol*) tem como principal função a tradução de endereços de forma dinâmica na rede. O uso mais comum é prover o processo da tradução de endereços IP e de endereços físicos (MAC address). O Comando **arp** permite gerenciar as tabelas de tradução de endereços e é bastante útil para descoberta de *hosts* e na administração de serviços de rede.

- Para listar a tabela ARP
  - **Exemplo 2.50** `arp -an`
- Para remover uma entrada na tabela ARP
  - **Exemplo 2.51** `arp -i eth0 -d 192.168.1.1`
- Para adicionar uma entrada na tabela ARP de forma estática
  - **Exemplo 2.52** `arp -s 192.168.1.1 00:cc:00:ff:ff:ee`

## 2.4 netstat

**Descrição 2.4** Comando para realizar análises e amostragens das conexões de rede, tabelas de roteamento e estatísticas gerais. O comando `ss` é a opção em distribuições mais atuais.

- Para mostrar todas as portas TCP e UDP em escuta e o processos responsáveis
  - **Exemplo 2.53** `netstat -tultp`
- Para listar a tabela de roteamento
  - **Exemplo 2.54** `netstat -rn`
- Para mostrar todas as conexões TCP abertas
  - **Exemplo 2.55** `netstat -tlnp`
- Para mostrar todas as conexões TCP abertas, de forma contínua
  - **Exemplo 2.56** `netstat -tlnpc`
- Para mostrar todas as conexões TCP abertas, modo estendido
  - **Exemplo 2.57** `netstat -tulpen`
- Para listar todas as conexões

- **Exemplo 2.58** netstat -a
- Para listar estatísticas por protocolo
  - **Exemplo 2.59** netstat -s
- Para listar estatísticas somente do protocolo UDP
  - **Exemplo 2.60** netstat -su
- Para listar estatísticas somente do protocolo TCP
  - **Exemplo 2.61** netstat -st
- Para listar estatísticas das interfaces
  - **Exemplo 2.62** netstat -i
- Para obter informações sobre os temporizadores das conexões
  - **Exemplo 2.63** netstat -o
- Para listar a tabela de roteamento IPv6
  - **Exemplo 2.64** netstat -rn -A inet6

## 2.5 route

**Descrição 2.5** Comando para mostrar informações e gerenciar tabelas de roteamento em IPv4 e IPv6. Este comando vem sendo substituído pelo **ip** em distribuições mais atuais.

- Para listar a tabela de roteamento
  - **Exemplo 2.65** route
- Para adicionar a rota padrão
  - **Exemplo 2.66** route add default gw 192.168.1.1
- Para adicionar uma rota estática
  - **Exemplo 2.67** route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0
  - **Exemplo 2.68** route add -net 192.168.1.0 netmask 255.255.255.0 gw 192.168.1.254
- Para remover rotas estáticas
  - **Exemplo 2.69** route del -net 192.168.1.0 netmask 255.255.255.0 gw 192.168.1.254
- Para remover a rota padrão
  - **Exemplo 2.70** route del default gw 192.168.1.1
- Para adicionar uma rota IPv6 padrão
  - **Exemplo 2.71** route -A inet6 add default gw 2001:0db8:0:200::1
- Para remover uma rota IPv6 padrão
  - **Exemplo 2.72** route -A inet6 del default gw 2001:0db8:0:200::1
- Para listar a tabela de roteamento IPv6
  - **Exemplo 2.73** route -A inet6

## 2.6 ethtool

**Descrição 2.6** Comando para gerenciar funções de mais baixo nível da interface de rede, tais como auto-negociação, velocidade, duplex e estatísticas de uso.

- Para mostrar informações gerais sobre a interface eth0
  - **Exemplo 2.74** `ethtool eth0`
- Para mostrar informações sobre o módulo do kernel (driver) da interface eth0
  - **Exemplo 2.75** `ethtool -i eth0`
- Para mostrar estatísticas de tráfego da interface eth0
  - **Exemplo 2.76** `ethtool -S eth0`
- Para mostrar informações sobre configurações de TX, RX e auto-negociação
  - **Exemplo 2.77** `ethtool -a eth0`
- Para fazer piscar o led da placa (ajudar na identificação física da placa)
  - **Exemplo 2.78** `ethtool -p eth0`
- Para forçar 100Mbit/s na interface eth0
  - **Exemplo 2.79** `ethtool -s eth0 speed 100`
- Para desabilitar a auto-negociação na interface eth0
  - **Exemplo 2.80** `ethtool -s eth0 autoneg off`
- Para forçar o modo full-duplex na interface eth0
  - **Exemplo 2.81** `ethtool -s eth0 duplex full`
- Para ativar a opção Wake-on-LAN na interface eth0
  - **Exemplo 2.82** `ethtool -s eth0 wol g`

## 2.7 dig

**Descrição 2.7** Comando para realizar consultas servidores DNS (*Domain Naming System*), alternativa aos comandos **nslookup** e **host**.

- Para descobrir informações sobre um domínio
  - **Exemplo 2.83** `dig www.senacrs.com.br`
- Para descobrir informações sobre um domínio, com saída reduzida
  - **Exemplo 2.84** `dig www.senacrs.com.br +noall +answer`
- Para descobrir informações sobre um domínio, com saída mínima
  - **Exemplo 2.85** `dig www.senacrs.com.br +short`
- Para descobrir informações sobre os servidores de e-mail de um domínio
  - **Exemplo 2.86** `dig MX senacrs.com.br`
- Para descobrir informações sobre os servidores de DNS de um domínio

- **Exemplo 2.87** dig NS senacrs.com.br
- Para receber informações sobre registros de IPv6 de um domínio
  - **Exemplo 2.88** dig AAAA senacrs.com.br
- Para receber o maior número de informações sobre um domínio
  - **Exemplo 2.89** dig ANY senacrs.com.br
- Para fazer consulta sobre o DNS reversos para determinado IP
  - **Exemplo 2.90** dig -x 177.1.214.233
- Para fazer consulta sobre o DNS reversos para determinado IP, com saída reduzida
  - **Exemplo 2.91** dig -x 177.1.214.233 +short
- Para descobrir informações sobre um domínio, consultando outro servidor, no exemplo o servidor 8.8.8.8
  - **Exemplo 2.92** dig @8.8.8.8 www.senacrs.com.br
- Para realizar uma consulta completa para determinado domínio, passando pelos servidores raiz
  - **Exemplo 2.93** dig www.senacrs.com.br +trace
- Para obter informações sobre o SOA (*Start of Authority*) de um domínio
  - **Exemplo 2.94** dig SOA senacrs.com.br
- Para obter informações sobre configurações e validações de SPF de um domínio
  - **Exemplo 2.95** dig TXT senacrs.com.br

## 2.8 host

**Descrição 2.8** Comando para realizar consultas servidores DNS (*Domain Naming System*), alternativa aos comandos **nslookup** e **dig**.

- Para descobrir informações sobre um domínio
  - **Exemplo 2.96** dig www.senacrs.com.br
- Para descobrir informações sobre um domínio
  - **Exemplo 2.97** host www.senacrs.com.br
- Para descobrir informações sobre um domínio, com saída aumentada
  - **Exemplo 2.98** host -v www.senacrs.com.br
- Para descobrir informações sobre os servidores de e-mail de um domínio
  - **Exemplo 2.99** host -t MX senacrs.com.br
- Para descobrir informações sobre os servidores de DNS de um domínio
  - **Exemplo 2.100** host -t NS senacrs.com.br
- Para descobrir informações sobre o IPv6 de um domínio
  - **Exemplo 2.101** host -t AAAA senacrs.com.br

- Para receber o maior número de informações sobre um domínio
  - **Exemplo 2.102** `host -t ANY senacrs.com.br`
- Para fazer consulta sobre o DNS reversos para determinado IP
  - **Exemplo 2.103** `host 177.1.214.233`
- Para descobrir informações sobre um domínio, consultando outro servidor, no exemplo o servidor 8.8.8.8
  - **Exemplo 2.104** `host www.senacrs.com.br 8.8.8.8`
- Para obter informações sobre o SOA (*Start of Authority*) de um domínio
  - **Exemplo 2.105** `host -t SOA senacrs.com.br`
- Para obter informações sobre configurações e validações de SPF de um domínio
  - **Exemplo 2.106** `host -t TXT senacrs.com.br`

## 2.9 nslookup

**Descrição 2.9** Comando para realizar consultas servidores DNS (*Domain Naming System*), alternativa aos comandos **host** e **dig**.

- Para descobrir informações sobre um domínio
  - **Exemplo 2.107** `nslookup www.senacrs.com.br`
- Para descobrir informações sobre os servidores de e-mail de um domínio
  - **Exemplo 2.108** `nslookup -query=MX senacrs.com.br`
- Para descobrir informações sobre os servidores de DNS de um domínio
  - **Exemplo 2.109** `nslookup -query=NS senacrs.com.br`
- Para receber o maior número de informações sobre um domínio
  - **Exemplo 2.110** `nslookup -query=ANY senacrs.com.br`
- Para receber informações sobre IPv6 de um domínio
  - **Exemplo 2.111** `nslookup -query=AAAA senacrs.com.br`
- Para fazer consulta sobre o DNS reversos para determinado IP
  - **Exemplo 2.112** `nslookup 177.1.214.233`
- Para descobrir informações sobre um domínio, consultando outro servidor, no exemplo o servidor 8.8.8.8
  - **Exemplo 2.113** `nslookup www.senacrs.com.br 8.8.8.8`
- Para obter informações sobre o SOA (*Start of Authority*) de um domínio
  - **Exemplo 2.114** `nslookup -query=SOA senacrs.com.br`
- Para obter informações sobre configurações e validações de SPF de um domínio
  - **Exemplo 2.115** `nslookup -query=TXT senacrs.com.br`

## 2.10 ping

**Descrição 2.10** Comando para realizar testes de conectividade e condições da rede. Funciona baseado no protocolo ICMP e possibilita obter diagnósticos de perdas, atraso e alcance entre hosts.

- Para realizar teste contínuo
  - **Exemplo 2.116** `ping www.senacrs.com.br`
- Para realizar teste com 20 pacotes
  - **Exemplo 2.117** `ping -c 20 www.senacrs.com.br`
- Para realizar teste com pacotes de 1000 Bytes
  - **Exemplo 2.118** `ping -s 1000 www.senacrs.com.br`
- Para realizar teste com pacotes enviados no intervalo de 0,5 segundos
  - **Exemplo 2.119** `ping -i 0.5 www.senacrs.com.br`
- Para realizar teste com pacotes no modo flood (inundação)
  - **Exemplo 2.120** `ping -f www.senacrs.com.br`
- Para utilizar IPv6
  - **Exemplo 2.121** `ping6 www.senacrs.com.br`

## 2.11 traceroute

**Descrição 2.11** Comando para realizar testes de conectividade, condições da rede e traçar o caminho (rota) entre uma origem e um destino. O **traceroute** é baseado no protocolo ICMP e no campo TTL (*Time to Live*) do protocolo IP que é decrementado a cada passagem por um roteador e ao chegar a zero retorna uma mensagem ICMP para a origem.

- Para realizar testes de rota básico para determinado destino
  - **Exemplo 2.122** `traceroute www.senacrs.com.br`
- Para realizar testes de rota básico para determinado destino, em IPv6
  - **Exemplo 2.123** `traceroute6 www.senacrs.com.br`
- Para realizar testes de rota básico para determinado destino, sem resolução de nomes (DNS)
  - **Exemplo 2.124** `traceroute -n www.senacrs.com.br`
- Para realizar testes de rota básico para determinado destino, com o protocolo ICMP ao invés do UDP padrão
  - **Exemplo 2.125** `traceroute -I www.senacrs.com.br`

## 2.12 iperf

**Descrição 2.12** Ferramenta para realizar testes de vazão entre dois ou mais hosts. O **iperf** permite a utilização do protocolo TCP e UDP para realizar os testes de vazão que tem como objetivo medir o desempenho da rede.

- Para colocar em modo servidor, com o protocolo TCP e a porta padrão 5001
  - **Exemplo 2.126** `iperf -s`
- Para colocar em modo servidor, com o protocolo TCP e a porta padrão 5001, com IPv6
  - **Exemplo 2.127** `iperf -s -V`
- Para executar o cliente, com o protocolo TCP, 10 segundos de teste e a porta padrão 5001
  - **Exemplo 2.128** `iperf -c IP_Servidor`
- Para executar o cliente, com o protocolo TCP, 10 segundos de teste e a porta padrão 5001, com IPv6
  - **Exemplo 2.129** `iperf -V -c IP_Servidor`
- Para executar o cliente, com o protocolo TCP, 30 segundos de teste, relatórios a cada 1s e a porta padrão 5001
  - **Exemplo 2.130** `iperf -c IP_Servidor -i 1 -t 30`
- Para colocar em modo servidor, com o protocolo TCP e a porta 15001
  - **Exemplo 2.131** `iperf -s -p 15001`
- Para colocar em modo servidor, com o protocolo UDP e a porta padrão 5001
  - **Exemplo 2.132** `iperf -s -u`
- Para executar o cliente, com o protocolo UDP, 30 segundos de teste, relatórios a cada 1s e a porta 15001
  - **Exemplo 2.133** `iperf -c IP_Servidor -i 1 -t 30 -u -p 15001`
- Para executar o cliente, com o protocolo TCP, 30 segundos de teste, relatórios a cada 1s, a porta padrão 5001 e com 10 conexões em paralelo
  - **Exemplo 2.134** `iperf -c IP_Servidor -i 1 -t 30 -P 10`

## 2.13 arping

**Descrição 2.13** Comando para realizar consultas por meio do protocolo ARP (*Address Resolution Protocol*), similar a ferramenta **ping**. O objetivo do uso é descobrir se um determinado endereço físico está ativo em rede.

- Para enviar requisições ARP para um host vizinho pela interface eth0. Caso o host que tenha o IP 192.168.1.1 esteja ativo na rede, haverá o retorno com a resposta e o tempo de latência.
  - **Exemplo 2.135** `arping -I eth0 192.168.1.1`
- Para procurar endereços IP duplicados. Este comando envia mensagens ARP solicitando quem teria o endereço físico do endereço IP 192.168.1.1. O IP 192.168.1.1 seria um endereço IP conhecido e se estaria buscando mais alguma interface com o mesmo IP. Em caso de IPs duplicados seriam informados os endereços físicos dos hosts.



■ **Exemplo 2.136** `arping -D -I eth0 192.168.1.1`

## 2.14 wget

**Descrição 2.14** É um cliente em linha de comando do protocolo HTTP e HTTPS. Permite que sejam feitos acessos a servidores HTTP e downloads de arquivos (uso mais comum).

- Para fazer o download de uma URL

■ **Exemplo 2.137** `wget http://192.168.200.3/arquivo.iso`

- Para fazer o download de uma URL que possua usuário e senha

■ **Exemplo 2.138** `wget --http-user=aluno --http-password=senha http://192.168.200.3/arquivo.iso`

- Para fazer o download de uma URL por meio de um proxy

■ **Exemplo 2.139** `wget -e use_proxy=yes -e http_proxy=192.168.200.253:8080 http://192.168.200.3/arquivo.iso`

- Exportar as variáveis do shell `http_proxy` e `https_proxy` para uso com web proxy.

■ **Exemplo 2.140** `export http_proxy="http://192.168.200.253:8080"`

`export https_proxy="http://192.168.200.253:8080"`

## 2.15 ssh e scp

**Descrição 2.15** Comandos para realizar acesso remoto e cópia de arquivos por meio do protocolo SSH. Estas ferramentas são fundamentais para o gerenciamento de servidores Linux.

- Para acessar um servidor SSH (192.168.200.3), na porta padrão, com o usuário aluno.

■ **Exemplo 2.141** `ssh aluno@192.168.200.3`

- Para acessar um servidor SSH (192.168.200.3), na porta padrão, com o usuário aluno e obter o modo de depuração.

■ **Exemplo 2.142** `ssh -vvvv aluno@192.168.200.3`

- Para acessar um servidor SSH (192.168.200.3), na porta 34000, com o usuário aluno.

■ **Exemplo 2.143** `ssh -p34000 aluno@192.168.200.3`

- Para acessar um servidor SSH (192.168.200.3), na porta padrão, com o usuário aluno e executar o comando `"dig @8.8.8.8 www.senacrs.com.br"`.

■ **Exemplo 2.144** `ssh aluno@192.168.200.3 "dig @8.8.8.8 www.senacrs.com.br"`

- Para copiar o diretório `/opt/arquivos` do servidor remoto para o diretório local `/var/opt`, com a porta padrão e com o usuário aluno, mantendo as permissões dos arquivos e diretórios remotos.

■ **Exemplo 2.145** `scp -p -r aluno@192.168.200.3:/opt/arquivos /var/opt`

- Para copiar o diretório local `/tmp/relatorio` para o servidor remoto no diretório `/home/aluno`, com a porta 34000 e com o usuário aluno, mantendo as permissões dos arquivos e diretórios locais.

■ **Exemplo 2.146** `scp -P34000 -p -r /tmp/relatorio aluno@192.168.200.3:`

## 2.16 ss

**Descrição 2.16** Comando para realizar análises e amostragens das conexões de rede, tabelas de roteamento e estatísticas gerais. O comando **netstat** é a opção em distribuições mais antigas.

- Para mostrar todas as portas TCP e UDP em escuta e o processos responsáveis.

■ **Exemplo 2.147** `ss -tupl`

- Para mostrar todas as conexões TCP abertas.

■ **Exemplo 2.148** `ss -t -a`

- Para mostrar todas as conexões UDP abertas.

■ **Exemplo 2.149** `ss -u -a`

- Para mostrar todas as conexões TCP abertas, de forma contínua a cada 5s.

■ **Exemplo 2.150** `watch -n 5 "ss -t -a"`

- Para listar todas as conexões.

■ **Exemplo 2.151** `watch -n 5 "ss -t -a"`

- Para listar todas as conexões.

■ **Exemplo 2.152** `ss -an`

- Para listar estatísticas por protocolo.

■ **Exemplo 2.153** `ss -sa`

- Para obter informações sobre os temporizadores das conexões.

■ **Exemplo 2.154** `ss -o`

## 2.17 mtr

**Descrição 2.17** Comando para realizar consultas a roteadores e traçar a rota entre dois hosts. O **mtr** é uma versão do utilitário **traceroute** com mais opções.

- Para realizar testes de forma contínua para determinado endereço.

■ **Exemplo 2.155** `mtr 8.8.8.8`

- Para realizar testes de forma contínua para determinado endereço, com IPv6.

■ **Exemplo 2.156** `mtr -6 www.google.com`

- Para realizar testes por 10 vezes e gerar um relatório.

■ **Exemplo 2.157** `mtr -r -c 10 8.8.8.8`

- Para realizar testes de forma contínua para determinado endereço, sem resolução de nomes (DNS).

■ **Exemplo 2.158** `mtr -n 8.8.8.8`

## 2.18 dhclient

**Descrição 2.18** Comando que implementa um cliente do protocolo DHCP e permite gerenciar o empréstimo de endereços IP.

- Para renovar (renew) o IP por DHCP na interface eth0.

■ **Exemplo 2.159** `dhclient eth0`

- Para liberar (release) o IP por DHCP na interface eth0.

■ **Exemplo 2.160** `dhclient -r eth0`

## 2.19 telnet

**Descrição 2.19** Comando que implementa um cliente do protocolo Telnet. É usado para realizar testes em portas de serviços que utilizam o protocolo TCP.

- Para fazer uma conexão a porta 80 de um endereço IP.

■ **Exemplo 2.161** `telnet 19.168.200.3 80`

- Obs.: para cancelar a conexão, utilizar a combinação de teclas CTRL+C ('^C')

## 2.20 netcat (nc)

**Descrição 2.20** Comando que provê diversas funcionalidades relacionadas a conexões de rede e para diagnóstico de serviços.

- Para criar um servidor, com o protocolo TCP, na porta 8000.

■ **Exemplo 2.162** `nc -l 8000`

- Para conectar em um servidor, com o protocolo TCP, na porta 8000.

■ **Exemplo 2.163** `nc 192.168.200.3 8000`

- Para criar um servidor, com o protocolo UDP, na porta 8000.

■ **Exemplo 2.164** `nc -u -l 8000`

- Para conectar em um servidor, com o protocolo UDP, na porta 8000.

■ **Exemplo 2.165** `nc -u 192.168.200.3 8000`

- Para criar um servidor, com o protocolo TCP, na porta 8000, e manter o socket aberto depois da primeira conexão.

■ **Exemplo 2.166** `nc -k -l 8000`

- Para transmitir um arquivo do lado do cliente para o lado do servidor.

■ **Exemplo 2.167** Servidor: `nc -l 8000 > /tmp/arquivo.dat`

■ **Exemplo 2.168** Cliente: `nc 192.168.200.3 8000 < arquivo.dat`

- Para transmitir um arquivo do lado do servidor para o lado do cliente.

■ **Exemplo 2.169** Servidor: `nc -l 8000 < /tmp/arquivo.dat`

■ **Exemplo 2.170** Cliente: `nc 192.168.200.3 8000 > arquivo.dat`

## 2.21 curl

**Descrição 2.21** É um cliente em linha de comando do protocolo HTTP e HTTPS. Permite que sejam feitos acessos a servidores HTTP e downloads de arquivos (uso mais comum).

- Realizar o download do arquivo.zip.
  - **Exemplo 2.171** `curl https://192.168.254.95/arquivo.zip`
- Realizar o download do arquivo.zip e salvar com o nome de arquivo.zip\_bk.
  - **Exemplo 2.172** `curl -o arquivo.zip_bk https://192.168.254.95/arquivo.zip`
- Realizar o download do arquivo.zip, com o limite de 1 Mbit/s.
  - **Exemplo 2.173** `curl --limit-rate 1m -O https://192.168.254.95/arquivo.zip`
- Realizar o download do arquivo "arquivo.zip" utilizando protocolo FTP no servidor 192.168.254.90, por meio dos dados de autenticação aluno e senha teste.
  - **Exemplo 2.174** `curl -u aluno:teste ftp://192.168.254.90/arquivo.zip`
- Realizar o download do arquivo "arquivo.zip" na URL `http://www.meusite.com.br` utilizando um proxy de endereço 192.168.254.1 na porta 3128.
  - **Exemplo 2.175** `curl -x 192.168.254.1:3128 http://www.meusite.com.br/arquivo.zip`
- Realizar o download do arquivo "arquivo.zip" utilizando uma conexão HTTPS e ignorando os erros de certificados SSL.
  - **Exemplo 2.176** `curl --insecure https://www.meusite.com.br/arquivo.zip`

## 2.22 rsync

**Descrição 2.22** Utilitário e protocolo para realizar a sincronização de arquivos e diretórios de forma local ou remota.

- Sincroniza arquivos locais .pdf em /dados para o host 192.168.254.29 em /backup utilizando o usuário root.
  - **Exemplo 2.177** `rsync -avz --include '*.pdf' /dados/ root@192.168.254.29:/backup`
- Sincroniza o diretório atual, mostrando o progresso da cópia e possibilitando o resumo da cópia em caso de falhas, limitando em 600 Kbytes/s para o host remoto 192.168.254.95 em /root/teste.
  - **Exemplo 2.178** `rsync -avh --partial --progress --bwlimit=600 . root@192.168.254.95:/root/teste`
- Sincroniza o diretório local, /dados/arquivos, com o diretório remoto /dados no host 192.168.254.95. O uso do rsync será por meio do protocolo SSH, com o usuário "aluno" no host remoto.
  - **Exemplo 2.179** `rsync -avz -e "ssh -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null" --progress /dados/arquivos/ aluno@192.168.254.95:/dados/`
- Obs.: o diretório arquivos será sincronizado como um subdiretório em /dados
  - **Exemplo 2.180** `rsync -avz -e "ssh -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null" --progress /dados/arquivos/* aluno@192.168.254.95:/dados/`

- Obs.: o conteúdo do diretório arquivos será sincronizado no diretório em /dados

## 2.23 iwconfig

**Descrição 2.23** Comando para realizar configurações na interface de rede sem fios (wireless) e obter estatísticas diversas.

- Para listar todas as interfaces sem fios disponíveis.
  - **Exemplo 2.181** `iwconfig -a`
- Para ativar a interface ath0.
  - **Exemplo 2.182** `iwconfig ath0 up`
- Para desativar a interface ath0.
  - **Exemplo 2.183** `iwconfig ath0 down`
- Para trocar para o canal 11 na interface ath0.
  - **Exemplo 2.184** `iwconfig ath0 interface channel 11`
- Para ativar o modo monitor na interface ath0.
  - **Exemplo 2.185** `iwconfig ath0 mode monitor`
- Para ativar o modo managed (padrão) na interface ath0.
  - **Exemplo 2.186** `iwconfig ath0 mode managed`

## 2.24 iwlist

**Descrição 2.24** Comando que provê funcionalidades de gerenciamento relacionadas a conexões de rede sem fios.

- Para listar os clientes associados ao access point.
  - **Exemplo 2.187** `iwlist peers`
- Para realizar varredura de canais (site survey) utilizando a interface ath0.
  - **Exemplo 2.188** `iwlist ath0 scan`
- Para listar os canais disponíveis.
  - **Exemplo 2.189** `iwlist channel`

## 2.25 ntpdate

**Descrição 2.25** Cliente do protocolo NTP (Network Time Protocol) para ajuste de dia e horário. Para o ajuste ser correto a zona de tempo (timezone) do host deverá estar correta. No horário de Brasília a zona de tempo é a GMT-3.

- Para ajustar o horário do sistema conforme o servidor de NTP 200.132.0.132.
  - **Exemplo 2.190** `ntpdate 200.132.0.132`
- Para apenas solicitar informações sobre dia e horário ao servidor NTP 200.132.0.132.
  - **Exemplo 2.191** `ntpdate -q 200.132.0.132`



## 3. Procedimentos

Neste capítulo são apresentados procedimentos comuns para configurações de rede no sistema operacional Linux.

### 3.1 Renomear interface

Para trocar o nome da interface **eth0** para **externo** em tempo de execução

#### Procedimento 3.1.1

```
ip link set eth0 down  
ip link set eth0 name externo  
ip link set externo up
```

Para trocar o nome da interface **eth0** para **externo** de forma permanente, editar o arquivo **/etc/udev/rules.d/70-persistent-net.rules** e trocar o nome da interface que corresponde ao endereço MAC:

#### Procedimento 3.1.2

```
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="*", ATTR{address}=="00:0c:29:1d:86:fd",  
ATTR{type}=="1", KERNEL=="eth*", NAME="eth0"  
  
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="*", ATTR{address}=="00:0c:29:1d:86:fd",  
ATTR{type}=="1", KERNEL=="eth*", NAME="externo"
```

### 3.2 Uso de VLANs

Para ativar o uso de VLANs no Linux, deve ser usado o módulo **8021q**. Por exemplo:

#### Procedimento 3.2.1

```
modprobe 8021q
```

Para criar uma interface virtual deve-se utilizar o comando **vconfig**. Por exemplo, para criar a interface **eth1.100** com a marcação 100

#### Procedimento 3.2.2

```
vconfig add eth1 100
```

Para evitar problemas com a filtragem de pacotes, deve-se ativar a **flag** para tornar o **dump** (uso em capturas de tráfego de rede) da interface tal como se não houvesse VLANs. Por exemplo:

**Procedimento 3.2.3**

```
vconfig set_flag eth1.100 1
```

Para remover uma VLAN, usa-se o comando **vconfig** com o parâmetro **rem**. Por exemplo, para remover a interface eth1.100:

**Procedimento 3.2.4**

```
vconfig rem eth1.100
```

Para definir um endereço IP de uma interface com VLAN, usa-se o comando **ifconfig** padrão. Por exemplo, para definir o IP 10.0.0.100 na interface eth1.100:

**Procedimento 3.2.5**

```
ifconfig eth1.100 10.0.0.100/24
```

### 3.3 Interface em Bridge

Para criar uma bridge de nome br0

**Procedimento 3.3.1**

```
ip link add br0 type bridge  
brctl addbr br0
```

Para adicionar a interface eth0 na bridge br0

**Procedimento 3.3.2**

```
brctl addif br0 eth0  
ip link set eth0 master br0
```

Para mostrar informações sobre a bridge br0

**Procedimento 3.3.3**

```
brctl show
```

Para mostrar a tabela MAC da bridge br0

**Procedimento 3.3.4**

```
brctl showmacs br0
```

Para ativar a bridge de nome br0

**Procedimento 3.3.5**

```
ip link set up dev br0
```

Para desativar a bridge de nome br0



**Procedimento 3.3.6**

```
ip link set dev br0 down
```

Para remover a interface eth0 de uma bridge de nome br0

**Procedimento 3.3.7**

```
ip link set eth0 nomaster
```

```
ip link set eth0 down
```

Para remover uma bridge de nome br0

**Procedimento 3.3.8**

```
ip link delete br0 type bridge
```

```
brctl delbr br0
```

Para configurar a interface bridge na inicialização

**Procedimento 3.3.9****Debian****Arquivo /etc/network/interfaces**

```
auto lo br0
```

```
iface lo inet loopback
```

```
iface eth0 inet manual
```

```
iface eth1 inet manual
```

```
# Bridge br0
```

```
iface br0 inet static
```

```
bridge_ports eth0 eth1
```

```
address 192.168.200.3
```

```
broadcast 192.168.200.255
```

```
netmask 255.255.255.0
```

```
gateway 192.168.200.1
```

**Procedimento 3.3.10****Centos****Arquivo /etc/sysconfig/network-scripts/ifcfg-br0**

```
DEVICE=br0
```

```
TYPE=Bridge
```

```
IPADDR=192.168.200.3
```

```
NETMASK=255.255.255.0
ONBOOT=yes
BOOTPROTO=none
NM_CONTROLLED=no
DELAY=0
```

**Arquivo /etc/sysconfig/network-scripts/ifcfg-eth0**

```
DEVICE=eth0
TYPE=Ethernet
HWADDR=AA:BB:CC:DD:EE:FF
BOOTPROTO=none
ONBOOT=yes
NM_CONTROLLED=no
BRIDGE=br0
```

**Arquivo /etc/sysconfig/network-scripts/ifcfg-eth1**

```
DEVICE=eth1
TYPE=Ethernet
HWADDR=AA:BB:CC:DD:EE:FE
BOOTPROTO=none
ONBOOT=yes
NM_CONTROLLED=no
BRIDGE=br0
```

**Obs.: é necessária a instalação do pacote bridge-utils para o utilitário brctl**

### 3.4 Interface em Bonding

Para definir qual o modo de operação da interface bonding de nome bond0, editar o arquivo **/etc/modprobe.d/bonding.conf**

**Procedimento 3.4.1**

```
alias bond0 bonding
options bond0 miimon=80 mode=0
Obs.: mode=1 (Active-Passive), mode=0 (Round-Robin)
```

Para listar sobre a interface bond0

**Procedimento 3.4.2**

```
cat /proc/net/bonding/bond0
```

Para criar a interface bond0, com as interfaces físicas eth0 e eth1, no modo round-robin

**Procedimento 3.4.3**

```
modprobe bonding  
ifenslave bond0 eth0 eth1  
ip link set bond0 up
```

Para trocar a interface ativa para eth1, no modo de operação Active-Passive

**Procedimento 3.4.4**

```
ifenslave -c bond0 eth1
```

Para remover a interface eth0 do bonding

**Procedimento 3.4.5**

```
ifenslave -d bond0 eth0
```

Ativação da interface bond0 na inicialização

**Procedimento 3.4.6****CentOS****Arquivo /etc/sysconfig/network-scripts/ifcfg-eth0**

```
DEVICE="eth0"  
BOOTPROTO="none"  
ONBOOT="yes"  
TYPE="Ethernet"  
MASTER=bond0  
SLAVE=yes
```

**Arquivo /etc/sysconfig/network-scripts/ifcfg-eth1**

```
DEVICE="eth1"  
BOOTPROTO="none"  
ONBOOT="yes"  
TYPE="Ethernet"  
MASTER=bond0
```

```
SLAVE=yes
```

**Arquivo /etc/sysconfig/network-scripts/ifcfg-bond0**

```
DEVICE=bond0
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.200.3
PREFIX=24
NETWORK=192.168.200.0
GATEWAY=192.168.200.1
```

**Procedimento 3.4.7****Debian****Arquivo /etc/network/interfaces**

```
auto bond0
iface bond0 inet static
address 192.168.200.3
netmask 255.255.255.0
network 192.168.200.0
gateway 192.168.200.1
slaves eth0 eth1
bond_mode active-backup
# bond_mode 0 para Round-Robin
bond_miimon 100
bond_downdelay 200
bond_updelay 200
```

### 3.5 Configurações de interface em arquivos

Configurações de interfaces de rede em sistemas padrão **Debian**.

**Procedimento 3.5.1****Arquivo de configuração: /etc/network/interfaces****# Exemplo de IPv4**

```
auto eth0
```

```
iface eth0 inet static
address 192.168.200.3
netmask 255.255.255.0
gateway 192.168.200.254
broadcast 192.168.200.255
dns-nameservers 192.168.200.1 8.8.8.8
dns-search empresa.local
```

### **# Exemplo de IPv6**

```
iface eth0 inet6 static
address fc00:0:2010:60::190
netmask 64
gateway fc00:0:2010:60::191
```

### **# Exemplo com o uso de VLANs**

#### **# Primeira opção: chamar um script para a criação das interfaces marcadas**

```
auto eth1
iface eth1 inet manual
up ifconfig eth1 0.0.0.0 up
up /root/vlans/vlan.sh
```

#### **# Segunda opção: adicionar as interfaces com a nomeação para uso de VLANs**

```
auto eth0.2
iface eth0.2 inet static
address 192.168.2.1
netmask 255.255.255.0
```

```
auto eth0.3
iface eth0.3 inet static
address 192.168.3.1
netmask 255.255.255.0
```

### **# IP alias**

```
auto eth0:1
```

```
iface eth0:1 inet static
address 192.168.1.7
netmask 255.255.255.0
broadcast 192.168.1.255
network 192.168.1.0
```

Configurações de interfaces de rede em sistemas padrão **Redhat/CentOS**).

#### Procedimento 3.5.2

**Arquivo de configuração: /etc/sysconfig/network-scripts/ifcfg-eth0**

**# Exemplo de IPv4 e IPv6**

```
DEVICE="eth0"
BOOTPROTO="static"
BROADCAST="192.168.200.255"
DNS1="8.8.8.8"
GATEWAY="192.168.200.254"
HWADDR="00:50:56:A8:6F:66"
IPADDR="192.168.200.3"
IPV6ADDR="fc00:0:2010:60::116/64"
IPV6INIT="yes"
IPV6_AUTOCONF="no"
NETMASK="255.255.255.0"
NM_CONTROLLED="yes"
ONBOOT="yes"
TYPE="Ethernet"
UUID="849a1bcf-9d10-4fca-a910-6b0e9af18aba"
IPV6_DEFAULTGW=fc00:0:2010:60::191
```

**# VLAN (um arquivo para cada interface)**

**# Arquivo de configuração: /etc/sysconfig/network-scripts/ifcfg-eth1.192**

```
DEVICE=eth1.192
BOOTPROTO=none
ONBOOT=yes
IPADDR=192.168.1.1
```

```
PREFIX=24
NETWORK=192.168.1.0
VLAN=yes

# IP alias
# Arquivo de configuração: /etc/sysconfig/network-scripts/ifcfg-eth1:0
DEVICE=eth1:0
BOOTPROTO=none
ONBOOT=yes
IPADDR=192.168.1.1
PREFIX=24
NETWORK=192.168.1.0
NAME=eth0:0
```

### 3.6 iftop

**Descrição 3.1** Comando para realizar o monitoramento do tráfego em interfaces de rede. Esta ferramenta permite o uso de filtros compatíveis com a biblioteca Libpcap (tcpdump).

#### Procedimento 3.6.1

Para monitorar o tráfego de rede na interface eth0.

■ **Exemplo 3.1** iftop -i eth0

Para monitorar o tráfego de rede na interface eth0 que seja relacionado ao endereço IP 192.168.254.254.

■ **Exemplo 3.2** iftop -i eth0 -f "host 192.168.254.254"

Para monitorar o tráfego de rede na interface eth2 que seja relacionado a rede 192.168.254.0/24 e na porta TCP 8080.

■ **Exemplo 3.3** iftop -i eth0 -f "net 192.168.254 and tcp port 8080"