

# Sistemas Operacionais

## Comandos Básicos do Linux

Felipe Augusto Lima Reis

[felipe.reis@ifmg.edu.br](mailto:felipe.reis@ifmg.edu.br)



**INSTITUTO  
FEDERAL**  
Minas Gerais

## Disclaimer

- Os comandos descritos neste documento foram feitos com base no sistema Linux Ubuntu 20.04;
  - Alguns comandos podem ser ligeiramente diferentes em outras distribuições;
  - A instalação de programas será baseada em pacotes Debian (.deb) - outras distribuições podem utilizar pacotes diferentes, como o RPM.

## Disclaimer

- Todos os comandos descritos neste documento devem ser executados usando letras minúsculas
  - Parâmetros, entretanto, podem utilizar letras maiúsculas - ver sintaxe dos comandos;
  - Os colchetes (“[.]”) não devem ser utilizados nos comandos - somente indicam que o usuário deverá substituir a informação existente dentro do colchetes;
    - A informação dentro do colchetes pode indicar um parâmetro, que fornece um comportamento específico do software, ou um nome de arquivo a ser aberto por um comando de escrita de arquivos.

# Sumário

- 1 Básicos
- 2 Gerenc. Programas
- 3 Diretórios
- 4 Arquivos
- 5 Permissão
- 6 Monitoramento
- 7 Redes e Segurança

# COMANDOS BÁSICOS

# SUDO

- **SUDO** possibilita a execução de comandos como um superusuário (root).
  - Também pode ser utilizado para execução de um comando como um usuário específico.

- Sintaxe:

`sudo [opções] [comando]`

- Opções comuns:
  - **-h**: exibe opções do comando;
  - **-u usuário**: executa o comando com privilégios do usuário especificado no comando.
  - **sudo su**: inicializa o modo superusuário - todos os comandos na sequência serão executados no modo superusuário.

---

As opções **-h** ou **-help** são frequentemente usadas para auxílio de um determinado comando no ambiente Linux.

# EXIT

- **EXIT** é o comando utilizado para fechar o terminal ou sair do modo *sudo*.
- Sintaxe:

`exit`

# CTRL+C e CTRL+Z

- **CTRL+C** é a tecla de atalho usada para interromper processos no terminal (caso o programa esteja em execução);
- **CTRL+Z** é a tecla de atalho usada para suspender a execução de um processo no terminal (caso o programa esteja em execução).
  - Quando CTRL+C não for capaz de finalizar um programa (travado no terminal), finalize se consegue retomar o controle do terminal usando CTRL+Z;
  - Em seguida, é possível finalizar o processo utilizando o comando `kill` (será detalhado a frente).

---

Para copiar textos no terminal utilize o atalho **CTRL+SHIFT+C**. Para colar, utilize **CTRL+SHIFT+V**.



# Manual de Comandos (Ajuda)

- **MAN** é o comando correspondente ao manual de comandos do sistema operacional.
  - Fornece um manual / tutorial básico do comando utilizado como parâmetro;
  - Indica opções e parâmetros existentes no comando;
  - Para navegar no manual, utilize as setas direcionais do teclado;
  - Para encerrar o comando, digite *q* (quit).
- Sintaxe:

`man [comando]`

# Desligamento do Sistema

- **SHUTDOWN** é o comando utilizado para desligar o sistema.
  - Ao executar o comando shutdown, alguns SOs agendam o desligamento para alguns poucos minutos;
  - Muitos sistemas operacionais exigem que o comando seja executado em modo sudo.
- Sintaxe:  

```
sudo shutdown [opções]
```
- Opções comuns:
  - **-r**: agenda reinício do sistema;
  - **now**<sup>1</sup>: desliga o sistema logo após a execução do comando;
  - **-r now**: reinicia o sistema logo após a execução do comando.

---

<sup>1</sup>Pode ser traduzido como agora.

# Desligamento, Reinício e Suspensão do Sistema

- **REBOOT**, **POWEROFF** E **HALT** são comandos utilizados para reiniciar, desligar e suspender o sistema
  - Podem ser utilizado em substituição ao comando `shutdown`.
- Sintaxe:

```
sudo reboot
```

---

A sintaxe dos comandos `poweroff` e `halt` é a mesma do comando `reboot`.

# INSTALAÇÃO, ATUALIZAÇÃO E REMOÇÃO DE PROGRAMAS

- **APT** é um conjunto de ferramentas em ambientes Debian (e derivações, incluindo Ubuntu) para administração de pacotes .deb de forma automática
  - O APT instala e/ou atualiza programas e dependências (pacotes necessários para ao funcionamento de um programa);
  - O APT acessa **repositórios** de programas (ou pacotes), gerenciados pelas distribuições (ou pelos desenvolvedores) e faz download de todos os arquivos necessários à instalação.

---

APT é acrônimo de *Advanced Packaging Tool* (Ferramenta de Empacotamento Avançada).

- **Snappy** ou **Snap** é um sistema de gerenciamento e administração de pacotes criado pela Canonical;
- Tem como objetivo ser uma tecnologia universal, comum a diferentes ambientes Linux
  - A loja de softwares dos sistemas Ubuntu instala programas utilizando essa tecnologia;
  - Os pacotes são chamados de **snaps** e a ferramenta utilizada para instalação destes é denominada **snapd**;
  - O snap funciona como um container, isolando o aplicativo do sistema, de modo que ele não interfira em nenhum outro [?].

---

O software Flatpak, usado em diversas distribuições Linux, é uma alternativa ao Snap.

# ATUALIZAÇÃO DO SISTEMA

# Atualização do Sistema

- **APT UPDATE** é o comando utilizado para atualizar a lista de pacotes a serem instalados no sistema.

```
sudo apt update
```

- **APT UPGRADE** é o comando utilizado para atualizar e instalar efetivamente pacotes no sistema.

```
sudo apt upgrade
```

---

Tradicionalmente era utilizada a sintaxe “sudo apt-get [comando]”. No entanto, nos sistema atuais, é possível utilizar somente “sudo apt [comando]”.



# Atualização do Sistema

- **APT DIST-UPGRADE** é o comando utilizado para atualizar, remover e instalar pacotes no sistema.
  - O `apt dist-upgrade` pode remover pacotes antigos para instalar novas versões, consistindo em um comando mais arriscado que o `apt upgrade`.

```
sudo apt dist-upgrade
```

- **APT CLEAN** é um comando utilizado para apagar arquivos baixados para uma instalação.

```
sudo apt clean
```

# INSTALAÇÃO E DESINSTALAÇÃO DE PROGRAMAS

## Instalação - via APT

- **APT INSTALL** é o comando utilizado para instalação de programas via APT.
  - Os programas (e as dependências) são baixados dos repositórios e instalados no computador;
  - É necessário o conhecimento do nome exato do programa / pacote - por muitas vezes essa busca deve ser feita de forma prévia na internet.
- Sintaxe:

```
sudo apt install [nome programa]
```

# Desinstalação - via APT

- **APT REMOVE**, **APT PURGE** e **APT AUTOREMOVE** são comandos utilizados para remoção de programas.
  - O **APT REMOVE** desinstala o software, mas mantém arquivos de configuração e dependências;
  - O **APT PURGE** desinstala o software e arquivos de configuração, mas mantém as dependências;
  - O **APT AUTOREMOVE** desinstala o software, arquivos de configuração e dependências, mantendo apenas dependências compartilhadas;
- Sintaxe:

```
sudo apt remove [nome programa]
```

---

A sintaxe dos comandos `apt purge` e `autoremove` é a mesma do comando `apt remove`.

# Instalação - via SNAP

- **SNAP INSTALL** é o comando utilizado para instalação de programas via SNAP.
  - Os programas (e suas dependências) são baixados e instalados no computador.
- Sintaxe:

```
sudo snap install [nome programa]
```

# Atualização e Desinstalação - via SNAP



- **SNAP REFRESH** é o comando utilizado para atualizar programas via SNAP.

```
sudo snap refresh [nome programa]
```

- **SNAP REMOVE** é o comando utilizado para remover programas via SNAP.

```
sudo snap remove [nome programa]
```

## Instalação - via DPKG

- **DPKG** é o comando utilizado para instalação de pacotes .deb (baixados via browser, por exemplo)
  - Utilizado para instalação de programas que não estão disponíveis nos repositórios;
  - Pode ser utilizado também para instalação de versões de softwares diferentes das existentes nos repositórios.
- Sintaxe:

```
sudo dpkg -i [arquivo_programa.deb]
```

---

A opção `dpkg -i` é utilizada para instalação. Outros parâmetros, como `-r` e `-P` são utilizados para desinstalação.

## Desinstalação - via DPKG

- **DPKG** pode ser utilizado também para desinstalação de pacotes .deb.

- O comando **dpkg -r** remove apenas o programa.

```
sudo dpkg -r [arquivo_programa.deb]
```

- O comando **dpkg -P** remove o programa e os arquivos de configuração.

```
sudo dpkg -P [arquivo_programa.deb]
```



# DIRETÓRIOS

# Listagem de Arquivos e Diretórios

- **LS** permite a listagem de arquivos e pastas em um diretório.
- Sintaxe:

```
ls [opções] [diretório]
```

- Opções comuns:
  - **-a**: exibe todos os arquivos, incluindo arquivos e pastas ocultos;
  - **-l**: exibe informações em lista, contendo detalhes de pastas e arquivos;
  - **-la**: exibe todos os arquivos (incluindo ocultos), em lista;
  - **-t**: exibe arquivos ordenados por data.

---

O parâmetro correspondente ao diretório não é obrigatório. Se não for indicado um diretório, o comando ls exibe informações do diretório corrente.

# Navegação

- **CD** permite navegar na árvore de diretórios do sistema.

- Sintaxe:

```
cd [diretório]
```

- Diretórios comuns:

- `..` : navega para o nó superior na árvore de diretórios;
- `/` : navega para o diretório raiz (/);
- `~` : navega para a pasta pessoal do usuário (home);

- Exemplo:

```
cd /home/ifmg/
```

# Criação de Diretórios

- **MKDIR** possibilita a criação de novos diretórios.

- Sintaxe:

```
mkdir [opções] [diretório]
```

- Opções comuns:

- **-p**: permite criar diretórios aninhados (*pasta1/pasta2/*);
- **-m [permissão]**: cria um diretório já estabelecendo as permissões de acesso (a=rwx).

- Exemplo:

```
mkdir aula_so
```

# Remoção de Diretórios

- **RMDIR** possibilita a remoção de diretórios.
- Sintaxe:  

```
rmmdir [opções] [diretório]
```
- Opções comuns:
  - **-p**: permite a remoção de diretórios e seus ancestrais.
- Exemplo:

```
rmmdir aula_so
```

# Mover Diretórios

- **MV** possibilita mover ou renomear diretórios.
  - Para mover um diretório, digite o nome do diretório e o caminho do novo diretório.
- Sintaxe:  

```
mv [dir. origem] [dir. destino]
```
- Exemplos:
  - Para pasta Desktop: `mv aula_so /home/Desktop`
  - Move para pasta superior: `mv aula_so ../`
  - Move para pasta atual: `mv /home/ifmg/aula_so .`

---

O comando **MV** é utilizado tanto para mover quanto para renomear diretórios.

# Copiar Diretórios

- **CP** possibilita copiar arquivos e diretórios.
  - É possível copiar múltiplos arquivos ou diretórios para um único diretório de destino;
  - Para cópia de diretórios é necessário utilizar o parâmetro `-r`.
- Sintaxe:  
`cp [opções] [arq. origem] [dir. destino]`
- Opções comuns:
  - `-r`: copia arquivos e diretórios recursivamente.
- Exemplos:
  - Copia diretório: `cp -r aula/ aula_bkp`

# Renomear Diretórios

- **MV** possibilita mover ou renomear diretórios.
  - Para renomear um diretório, digite o nome/caminho do diretório e, em seguida, o novo nome do diretório;
  - Para renomear, o diretório de destino deve ser o mesmo do diretório de origem.

- Sintaxe:

```
mv [dir. origem] [dir. destino]
```

- Exemplos:

- Muda nome diretório: `mv aula_so aula_so2`
- Move nome diretório não local (mesmo destino):  
`mv home/ifmg/aula_so home/ifmg/aula_so2`

---

O comando **MV** é utilizado tanto para mover quanto para renomear diretórios.



# Imprimir Diretório Corrente

- **PWD** imprime o endereço completo do diretório corrente.
  - Esse comando é útil em algumas distribuições Linux que não exibem o diretório corrente no terminal;
  - Também pode ser útil para exibir informações de diretórios referenciados por links simbólicos (atalhos).
- Sintaxe:  

```
pwd [opções]
```
- Opções comuns:
  - **-l**: exibe o diretório mesmo que contenha links simbólicos.

---

PWD é o acrônimo de Print Working Directory (Imprimir Diretório Corrente).

# ARQUIVOS

# CRIAÇÃO E VISUALIZAÇÃO DE ARQUIVOS

# Criar Arquivos

- **ECHO** é o utilizado para adicionar textos em arquivos.
  - Caso arquivo não exista, ele é criado.
  - Caso ele exista, ele pode ser sobrescrevido ou ser atualizado (dependendo dos parâmetros).
- Sintaxe:

```
echo "texto entre aspas" > [arquivo]
echo "texto entre aspas" >> [arquivo]
```
- Opções comuns:
  - **>>**: atualiza o arquivo, concatenando o texto no conteúdo (também cria arquivos com o texto no conteúdo);
  - **>**: sobrescreve o arquivo (apaga o anterior e cria um novo com o texto no conteúdo).

# Exibir Conteúdo de Arquivos (CAT)

- **CAT** é o comando utilizado para concatenar ou visualizar o conteúdo de arquivos.
- Sintaxe:  

```
cat [arquivo]
```
- Opções comuns:
  - **-n**: exibe o número das linhas.

---

O comando **CAT** é utilizado tanto para concatenar quanto visualizar o conteúdo de arquivos. Nesta seção, será indicado o comando com o objetivo de visualizar o conteúdo dos arquivos.

# Exibir Conteúdo de Arquivos (HEAD)

- **HEAD** é o comando utilizado para visualizar as 10 primeiras linhas de um arquivo.
  - É possível visualizar o conteúdo de mais de um arquivo.
- Sintaxe:

```
head [opções] [arquivo]
```
- Opções comuns:
  - **-n**: exibe o conteúdo das *n* primeiras linhas do arquivo.

# Exibir Conteúdo de Arquivos (TAIL)

- **TAIL** é o comando utilizado para visualizar as 10 últimas linhas de um arquivo.
  - É possível visualizar o conteúdo de mais de um arquivo.
- Sintaxe:  

```
tail [opções] [arquivo]
```
- Opções comuns:
  - **-n**: exibe o conteúdo das *n* últimas linhas do arquivo.

# Exibir Conteúdo de Arquivos (LESS)

- **LESS** é o comando utilizado para visualizar o conteúdo completo dos arquivos.
  - Para navegar no conteúdo, utilize as setas direcionais do teclado;
  - Para encerrar, digite *q* (quit).
- Sintaxe:

```
less [opções] [arquivo]
```



## Editar Arquivos (NANO)

- **NANO** é o comando utilizado para editar e visualizar o conteúdo dos arquivos.
  - O NANO pode ser considerado um editor de texto para ser utilizado no terminal;
  - As principais opções do editor são disponibilizadas na parte de baixo do programa;
  - Para navegar no conteúdo dos arquivos, utilize as setas direcionais do teclado.
- Sintaxe:

`nano [opções] [arquivo]`

# Editar Arquivos (NANO)

- **NANO** é o comando utilizado para editar e visualizar o conteúdo dos arquivos.

- Sintaxe:

`nano [opções] [arquivo]`

- Opções principais (dentro do software):
  - **CTRL+O**: salva o arquivo;
  - **CTRL+X**: finaliza o programa;
  - **CTRL+W**: pesquisa um conteúdo no arquivo;
  - **CTRL+R**: permite leitura do arquivo;
  - **CTRL+C**: cancela uma ação.

---

No programa as opções estão indicados com o sinal circunflexo (^), indicando o uso da tecla CTRL.

## RENOMEAR, COPIAR E REMOVER

# Mover Arquivos

- **MV** possibilita mover ou renomear arquivos.
  - Para mover um arquivo, digite o nome do arquivo e o caminho do novo arquivo.
- Sintaxe:
 

```
mv [arq. origem] [arq. destino]
```
- Exemplos:
  - Para pasta Desktop: `mv aula.txt ~/aula.txt`
  - Move para pasta superior: `mv aula.txt ../`
  - Move para pasta atual: `mv /home/ifmg/aula.txt .`

---

O comando **MV** é utilizado tanto para mover quanto para renomear arquivos.

# Renomear Arquivos

- **MV** possibilita mover ou renomear arquivos.
  - Para renomear um arquivo, digite o nome/caminho do arquivo e, em seguida, o novo nome do arquivo;
  - Para renomear, o diretório de destino deve ser o mesmo diretório do arquivo de origem.
- Sintaxe:
 

```
mv [arq. origem] [arq. destino]
```
- Exemplos:
  - Muda nome arquivo: `mv aula.txt aula_nova.txt`

---

O comando **MV** é utilizado tanto para mover quanto para renomear arquivos.

# Copiar Arquivos

- **CP** possibilita copiar arquivos e diretórios.
  - É possível copiar múltiplos arquivos ou diretórios para um único diretório de destino.
- Sintaxe:
 

```
cp [opções] [arq. origem] [dir. destino]
```
- Opções comuns:
  - **-r**: copia arquivos e diretórios recursivamente.
- Exemplos:
  - Duplica arquivo: `cp aula.txt aula2.txt`
  - Copia p/ novo diretório: `cp aula.txt aulas/aula.txt`

# Remover Arquivos

- **RM** possibilita a remoção de arquivos e diretórios.
- Sintaxe:  

```
rm [opções] [arquivo / diretório]
```
- Opções comuns:
  - **-r**: permite a remoção de arquivos, diretórios e seus ancestrais.
- Exemplo:

```
rm aula.txt
```

# Informações sobre os Arquivos

- **STAT** exibe informações sobre um determinado arquivo.
  - Exibe informações como o tamanho do arquivo, quantidade de blocos (de HD/SSD/etc) utilizados, permissões de acesso e datas de acesso/modificação.

- Sintaxe:

```
stat [opções] [arquivo]
```

- Exemplo:

```
stat aula.txt
```



# Alterar Horário de Acesso e Modificação

- **TOUCH** é o comando utilizado para alterar horários de acesso e modificação de um arquivo (timestamp).
  - Também pode ser utilizado para criar arquivos em branco (caso o arquivo não exista).
- Sintaxe:  

```
touch [opções] [arquivo]
```
- Opções comuns:
  - **-a**: modifica somente o horário de acesso ao arquivo;
  - **-m**: modifica somente o horário de modificação do arquivo;
  - **-time=WORD**: modifica o horário para uma data específica.

# PERMISSÃO DE ACESSO

# Permissão (CHOWN)

- **CHOWN** é o comando utilizado para o dono ou o grupo de um arquivo
- Sintaxe:
 

```
chown [opções] [usuário / grupo] [arquivo / dir.]
```
- Opções comuns:
  - **-R**: modifica o proprietário de arquivos em um diretório de forma recursiva;
- Exemplos:

```
chown ifmg arquivo.txt
chown ifmg:ifmg arquivo.txt
chown -R ifmg diretorio/
```

# Permissão (CHMOD)

- **CHMOD** é o comando utilizado para mudar as permissões de arquivos ou diretórios.

- Sintaxe:

```
chmod [opções] [permissão] [arquivo / dir.]
```

- Opções comuns:

- **-R**: modifica a permissão de acesso a arquivos de forma recursiva em um determinado diretório;
- **+x**: adiciona permissão para executar um arquivo;
- **+x**: remove permissão para executar um arquivo.

- Exemplos:

```
chmod 777 arquivo.txt
```

```
chown +x arquivo.o
```

# MONITORAMENTO DO SISTEMA

# Monitoramento de Memória (FREE)

- **FREE** é o comando utilizado verificar a quantidade de memória existente e usada no sistema.

- Sintaxe:

`free [opções]`

- Opções comuns:

- **-m**: exibe quantidade de memória em MB;
- **-g**: exibe quantidade de memória em GB;
- **-h**: formato para "humanos" (ajusta automaticamente);
- **-s**: atualiza uso de memória a cada *n* segundos.

# Monitoramento de CPU e Memória (TOP)

- **TOP** é um comando utilizado iniciar um aplicativo de monitoramento de CPU e memória.

- Sintaxe:

`top [opções]`

- Opções comuns:

- **-u**: filtra processos de um usuário específico;
- **-i**: exibe somente processos em execução (não *idle*);
- **-s**: alterar o intervalo de atualização;
- **-f**: adiciona ou remove colunas;
- **-k**: possibilita a finalização (*kill*) de processos;
- **-E**: exibe memória em diferentes unidades, como MB, GB, etc.

---

Ao digitar a tecla correspondente ao parâmetro durante a execução do programa, filtra-se o conteúdo.

# Monitoramento de CPU e Memória (HTOP)

- **HTOP** é um programa utilizado para iniciar um aplicativo de monitoramento de CPU e memória
  - Em alguns sistemas Linux, esse programa precisa ser instalado;
  - Em outros, como o Ubuntu Server, o HTOP já vem instalado.
- Sintaxe:  

```
htop [opções]
```
- Opções comuns:
  - **-e**: exibe informações sobre os processos;
  - **-t**: exibe a árvore de processos;
  - **-u**: permite filtrar processos de um único usuário;

---

No `htop`, é possível acessar diversas funcionalidades da aplicação a partir de um menu na parte inferior do programa. Ao digitar a tecla correspondente ao parâmetro durante a execução do programa, filtra-se o conteúdo.



# Monitoramento de HD/SSD (DF)

- **DF** é o comando utilizado para visualizar o uso das partições do HD
  - Ao indicar um diretório como parâmetro, o DF irá retornar o uso da partição onde o diretório está localizado;
  - A opção de diretório é útil quando as pastas estiverem localizadas em partições diferentes (ex.: /home).
- Sintaxe:
 

```
df [opções] [diretório]
```
- Opções comuns:
  - **-h**: formato de leitura para "humanos" (exibe uso de memória em MB e GB);

# Monitoramento de HD/SSD (DU)

- **DU** é o comando utilizado para estimar o uso de memória de um dado diretório
  - Ao listar um diretório, o DU exibe informações de todos os subdiretórios (recomenda-se filtrar com opções `-d` ou `-s`);
  - É possível utilizar estimar o uso de 2 ou mais diretórios.
- Sintaxe:
 

```
du [opções] [diretório]
```
- Opções comuns:
  - `-h`: formato de leitura para "humanos" (MB e GB);
  - `-d`: filtra o nível de profundidade para exibição de subdiretórios;
  - `-c`: exibe o total (útil para 2 ou mais diretórios);
  - `-s`: exibe um resumo (não exibe subdiretórios).

# Encerramento de Processos (KILL)

- **KILL** é um comando utilizado enviar um sinal de desligamento para um processo
  - Por padrão utiliza o sinal SIGTERM;
  - Caso seja utilizada a opção SIGKILL, o programa será finalizado abruptamente.
- Sintaxe:  

```
kill [opções] [PID]
```
- Opções comuns:
  - **-9**: denominado SIGKILL, encerra o processo abruptamente;
  - **-15**: denominado SIGTERM, é conhecido como *soft kill*, pois pode ser ignorado pelo programa.

# Encerramento de Processos (KILLALL)

- **KILLALL** é um comando utilizado enviar um sinal de desligamento para um processo
  - Por padrão utiliza o sinal SIGTERM;
- Sintaxe:
 

```
killall [opções] [nome processo]
```
- Opções comuns:
  - **-9**: denominado SIGKILL, encerra o processo abruptamente;
  - **-15**: denominado SIGTERM, é conhecido como *soft kill*, pois pode ser ignorado pelo programa;
  - **-o**: encerra processos mais velhos que um dado horário;
  - **-y**: encerra processos mais novos que um dado horário;
  - **-u**: encerra processos somente de um dado usuário.

# REDES, SEGURANÇA E DOWNLOAD

# MONITORAMENTO DE REDES

# Gerenciamento de IP (IFCONFIG)

- **IFCONFIG** é o comando utilizado para listar, configurar e controlar informações de interfaces de rede.
  - Para configuração, pode ser necessário utilizar o comando `sudo`.
- Sintaxe:
 

```
ifconfig [opções] [interface]
```
- Opções comuns:
  - **-a**: exibe todas as interfaces, ativas e inativas;
  - **-s**: exibe um resumo das interfaces.

---

## Informações extras:

- **lo**: denominado *loopback*, corresponde ao endereço IP para acesso ao próprio computador (*localhost*);
- **RX**: quantidade de pacotes recebidos;
- **TX**: quantidade de pacotes transmitidos.

# Gerenciamento de IP (IFUP e IFDOWN)



- **IFUP** e **IFDOWN** são, respectivamente, comandos utilizados para ligar e desligar interfaces de rede.
- Sintaxe:

```
ifup [interface]  
ifdown [interface]
```



# Testar Conectividade de Equipamentos (PING)

- **PING** é o comando utilizado para testar a conectividade de equipamentos (ex. servidores)
  - Usado para verificar se os destinatários estão ativos na rede ou na internet;
  - Usado também para medir o tempo de envio e resposta de pacotes, a fim de analisar problemas de velocidade na rede;
  - Utiliza o protocolo ICMP (*Internet Control Message Protocol*).
- Sintaxe:

```
ping [opções] [equip. destino]
```

# Atualização Interface Rede (DHCLIENT)

- **DHCLIENT** é o comando utilizado para forçar a interface de rede a obter um novo IP no servidor DHCP (ou no roteador)
  - DHCP (*Dynamic Host Configuration Protocol*) é um protocolo TCP/IP que capaz de conceder endereços IP.
- Sintaxe:

```
sudo dhclient [opções]
```

# Rotas Entre Servidores (TRACEROUTE)

- **TRACEROUTE** é uma aplicação utilizada para verificar as rotas percorridas entre o computador atual e o servidor / equipamento de destino
  - Pode ser utilizado para identificar saltos e tempos de respostas em redes LAN, devido ao uso de equipamentos (*hubs*, *switches*, roteadores) em cascata.
- Sintaxe:

```
traceroute [opções] [IP / domínio]
```

---

Em alguns sistemas, o traceroute não vem instalado por padrão.

# Rotas Entre Servidores (TRACEPATH)

- **TRACEPATH** é uma aplicação utilizada para verificar as rotas percorridas e definir o MTU entre o computador atual e o servidor / equipamento de destino
  - A Unidade Máxima de Transmissão (MTU) corresponde ao tamanho do maior pacote que uma camada de um protocolo de comunicação pode transmitir;
  - Essa métrica influencia na velocidade e qualidade de uma conexão.
- Sintaxe:

```
tracpath [opções] [IP / domínio]
```

---

Em alguns sistemas, o tracpath não vem instalado por padrão.

# SEGURANÇA

# Análise de Portas Abertas (NETSTAT)

- **NETSTAT** é uma aplicação utilizada para verificar quais as portas abertas em um servidor / computador.
  - Indica as portas que estão ativas e quais aplicações estão utilizando essas portas
- Sintaxe:
 

```
netstat [opções]
```
- Opções comuns:
  - **-t**: exibe portas abertas no protocolo TCP;
  - **-u**: exibe portas abertas no protocolo UDP;
  - **-p**: exibe programas que gerenciam as portas abertas;
  - **-a**: exibe todas as portas;
  - **-n**: exibe o número da porta.

Frequentemente é utilizada a sintaxe `netstat -tupan`, para trazer as principais informações.

## DOWNLOAD DE ARQUIVOS

# Download de Arquivos (WGET)

- **WGET** é o comando utilizado para fazer download de arquivos na internet.
  - O comando deve acessar uma url específica;
  - Pode ser utilizado para obter páginas web recursivamente, sendo uma opção para construção de *web crawlers*.
- Sintaxe:  

```
wget [url]
```
- Opções comuns:
  - **-r**: recursivo: entra em links e faz download das páginas;
  - **-l**: define o nível de profundidade da opção recursiva;
  - **-U**: simula o comportamento de um browser específico