

# Bíblia do usuário Linux

Escrito por Mark Security

Livro dedicado a iniciantes e aprendizes de Linux.

Criado em 03/07/2021

Um pouco sobre **Mark Security**.

**Estudante de Segurança da Informação, Desenvolvedor Web e de Aplicativos de Desktop, além de possuir breve conhecimento em área Forense e Análise de Malware (Engenharia Reversa).**

**Todos os créditos direcionado para Mark Security.**

Todo conteúdo escrito aqui provém de um conhecimento baseado em **Distribuições Linux** baseadas em **Ubuntu** e **Debian**, portanto é aconselhado que utilize algumas das **Distribuições** ou variantes delas.

## Capítulo 1 - Introdução

- \* Introdução ao Linux
- \* Sistema Operacional
- \* O que é Linux?
- \* Como surgiu o Linux?
- \* Primeiros passos
- \* Sistemas Linux para Pentest

## Capítulo 2 - Comandos básicos

- \* Entendendo repositório Linux
- \* Comandos básicos do Linux
- \* Criando e Removendo Diretórios

## Iniciando Formação sobre Linux

### Introdução ao Linux.

O capítulo Introdução contém explicações teóricas sobre o sistema operacional, você pode pular este capítulo caso já conheça estas explicações. Para quem está começando, muita teoria pode atrapalhar o aprendizado, é mais produtivo ver na prática o que o computador faz e depois porque ele faz isto. Mesmo assim, é recomendado ler estes capítulos pois seu conteúdo pode ser útil. Seguem abaixo

algumas dicas para um bom começo: Faça a leitura desta apostila e pratique o que aprendeu imediatamente. Isto facilita o entendimento do programa/comando. É preciso ter interesse em aprender, se você tiver vontade em aprender algo, você terá menos dificuldade do que em algo que não gosta e está se obrigando a aprender.

Decorar não adianta, pelo contrário, só atrapalha no aprendizado. Você precisa entender o que o comando faz, deste modo você estará também usando e desenvolvendo sua interpretação, e entenderá melhor o assunto. Curiosidade também é importante. Você talvez possa estar procurando um comando que mostre os arquivos que contém um certo texto, e isto fará você chegar até o comando **grep**, depois você conhecerá suas opções, etc.

Não desanime vendo outras pessoas que sabem mais que você, lembre-se que ninguém nasce sabendo. Uma pessoa pode ter mais experiência em um assunto no sistema como compilação de programas, configuração, etc, e você pode ter mais interesse em redes.

Ninguém pode saber tudo da noite para o dia, não procure saber tudo sobre o sistema de uma só vez senão não entenderá NADA. Caso tenha dúvidas sobre o sistema, procure ler novamente esta apostila, e caso ainda não tenha entendido procure ajuda nas milhares de páginas web disponíveis com documentação para o Linux. A interface gráfica existe, mas os melhores recursos e flexibilidade estão na linha de comando. Você pode ter certeza que o aprendizado no Linux ajudará a ter sucesso e menos dificuldades em usar qualquer outro sistema operacional.

## **Boa Sorte e bem vindo ao Linux!**

### **1.1 Sistema Operacional**

O **Sistema Operacional** é a interface ao usuário e seus programas com o computador. Ele é responsável pelo gerenciamento de recursos e periféricos (como memória, discos, arquivos, impressoras, **CD-ROMs**, etc) e a execução de programas. No Linux o **Kernel** é o Sistema Operacional. Você poderá construí-lo de acordo com a configuração de seu computador e os periféricos que possui.

### **1.2 O que é Linux?**

O Linux é um **sistema operacional** criado em 1991 por Linus Torvalds na universidade de **Helsinki** na Finlândia. É um sistema Operacional de código aberto distribuído gratuitamente pela Internet. Seu código fonte é liberado como Free Software (software gratuito) o aviso de copyright do kernel feito por Linus descreve detalhadamente isto e mesmo ele está proibido de fazer a comercialização do sistema.

Isto quer dizer que você não precisa pagar nada para usar o Linux, e não é crime fazer cópias para instalar em outros computadores, nós inclusive incentivamos você a fazer isto. Ser um sistema de código aberto pode explicar a performance, estabilidade e velocidade em que novos recursos são adicionados ao sistema.

### **Algumas características do sistema Linux:**

\* É de graça e desenvolvido voluntariamente por programadores experientes, hackers, e contribuidores espalhados ao redor do mundo que tem como objetivo a contribuição para a melhoria e crescimento deste sistema operacional.

\* Convivem sem nenhum tipo de conflito com outros sistemas operacionais (com o DOS, Windows, OS/2) no mesmo computador.

\* Multitarefa real

\* Multiusuário

\* Suporte a nomes extensos de arquivos e diretórios (255 caracteres)

\* Conectividade com outros tipos de plataformas como **Apple, Sun, Macintosh, Sparc, Alpha, PowerPc, ARM, Unix, Windows, DOS**, etc.

\* Utiliza permissões de acesso a arquivos, diretórios e programas em execução na memória RAM.

\* **NÃO EXISTEM VIRUS NO LINUX!** Desde 1991, nunca foi registrado NENHUM tipo de vírus para este sistema. Isto tudo devido a grande segurança oferecida pelas permissões de acesso do sistema que funcionam inclusive durante a execução de programas.

\* O sistema de arquivos usado pelo Linux (Ext3) organiza os arquivos de forma inteligente evitando a fragmentação e fazendo-o um poderoso sistema para aplicações multi-usuárias exigentes e gravações intensivas.

\* Suporte a diversos dispositivos e periféricos disponíveis no mercado, tanto os novos como obsoletos.

\* Entre muitas outras características que você descobrirá durante o uso do sistema.

Outro ponto em que ele se destaca é o suporte que oferece a placas, **CD-Roms** e outros tipos de dispositivos de última geração e mais antigos (a maioria deles já ultrapassados e sendo completamente suportados pelo sistema operacional). Este é um ponto forte para empresas que desejam manter seus micros em funcionamento e pretendem investir em avanços tecnológicos com as máquinas que possui.

Hoje o Linux é desenvolvido por milhares de pessoas espalhadas pelo mundo, cada uma fazendo sua contribuição ou mantendo alguma parte do **Kernel** gratuitamente. **Linus Torvalds** ainda trabalha em seu desenvolvimento e também ajuda na coordenação entre os desenvolvedores.

## Primeiros passo

**Este capítulo apresenta o Linux ao aluno. Serão vistos os conceitos de:**

- \* Usuário
- \* Grupo
- \* Superusuário
- \* Login (entrando no sistema)
- \* Logout (saindo do sistema)
- \* Shutdown/halt (desligando corretamente o sistema)

### Usuário

Como o Linux foi concebido para que várias pessoas pudessem utilizar os mesmos recursos presentes em uma única máquina, surgiu o conceito de usuário para diferenciar o que cada pessoa estivesse fazendo e quais recursos ela estivesse ocupando. O usuário é a identificação da pessoa que irá utilizar o sistema.

A identificação do usuário é feita por um “número de identificação” ou id, que é atribuído ao usuário durante a criação de sua conta no sistema.

Com a finalidade de garantir a integridade do trabalho de cada usuário, impedindo que um usuário altere o trabalho de outro, no momento de entrada no sistema, você deve informar a senha do seu usuário. O nome de usuário associado à senha é a sua “chave de entrada” no sistema, portanto deve ser guardada com cuidado.

## Grupos

O Linux também possui o conceito de “grupo”. Um grupo é, como o próprio nome diz, um agrupamento de vários usuários que devem compartilhar algumas características em comum como, por exemplo, permissões de acessos a arquivos e dispositivos.

## Superusuário

O superusuário é aquele que tem plenos poderes dentro do Linux. É o superusuário quem pode criar novos usuários, alterar direitos, configurar e fazer a atualização do sistema. Somente ele tem direito de executar essas atividades.

É recomendado utilizar a conta de superusuário somente quando for necessário configurar algo no sistema. Ainda assim é recomendado utilizá-la o mínimo possível para evitar que algum erro danifique o sistema.

## Entrando e Saindo do Sistema

Ao iniciar o Linux, um prompt semelhante ao ilustrado a seguir será mostrado:

```
Kali Linux (tty1)
Mark login:
```

Informe o seu login/nome de usuário. A seguir será solicitada a senha (Password) do usuário. Digite a senha do seu usuário.

Após informar o nome de usuário e a senha corretamente, você será levado ao prompt do sistema:

```
[mark@kali linux]$
```

OBS.: O linux tem terminais virtuais. Você pode alterar entre eles utilizando as teclas Alt-Fn, onde n pode variar de 1 até 6 na configuração padrão.

Pode-se utilizar o comando logout na linha de comando para se desconectar do sistema:

## Desligando o Sistema

A fim de evitar danos ao sistema de arquivos, é necessário que o superusuário pare o sistema antes de desligar o computador. Um dos comandos que podem ser utilizados é o comando **shutdown**. Este comando permite tanto desligar quanto reiniciar o computador.

```
[mark@kali linux]$ shutdown -h now
```

O comando acima permite desligar o computador imediatamente, enviando uma mensagem a todos os usuários que estão utilizando o sistema.

```
[mark@kali linux]$ shutdown -h -t 30 "Atenção: O sistema será desligado dentro de 30 segundos"
```

O comando acima finaliza todos os processos e desliga o computador dentro de 30 segundos, enviando a mensagem de aviso a todos os usuários logados no sistema.

O comando **halt** diz ao sistema que ele deverá desligar imediatamente.

```
[mark@kali linux]$ halt
```

Para reinicializar o sistema, pode-se utilizar, além do comando **shutdown**, o comando **reboot**:

```
[mark@kali linux]$ shutdown -r -t 30 "Atenção: O sistema será reiniciado dentro de 30 segundos"
```

Esta opção finaliza todos os processos e **reinicia** o computador após 30 segundos.

```
[mark@kali linux]$ reboot
```

O comando **reboot** chama o comando **shutdown** e ao final deste reinicia o sistema. Após executar os comandos deve-se aguardar até que o sistema esteja parado (**com a mensagem o sistema está parado ou Power down**) para então poder desligar seu computador ou esperar que ele reinicie.

## Sistemas Linux para Pentest

O sistema Linux é um sistema perfeito para se realizar operações de Testes de Intrusão, Análise de Malware, Desenvolvimento de aplicações, Análises Forenses e dentre outros tipos de práticas. Dentre muitos anos, o Linux tem sido uns dos melhores e se não a melhor opção para muitos praticantes das áreas ditas acima. Mas o por que? Bom, isso se dá ao fato da facilidade de utilizar o sistema e principalmente das ferramentas, que por muitas vezes são desenvolvidas especificamente para distribuições **Linux**.

A facilidade da utilização do **Linux** ocorre quando nós já possuímos a certeza de que aprendemos ao menos utilizar as funções necessárias do dia a dia, como atualização de repositórios, instalação de ferramentas, utilização de comandos a nosso favor e dentre outras infinitudes de situações.

## Capítulo 2

### Entendendo o repositório Linux

De forma mais simplificada de entender o que são repositórios, digo que são apenas servidores que hospedam os pacotes de softwares de cada distribuição Linux, isso me refiro a todo conteúdo que você consegue instalar no seu computador através do terminal ou de lojas de aplicativos.

## Atualizando repositórios

A atualização de repositórios é basicamente um meio de buscar os mais recentes softwares de cada servidor que hospedam os repositórios. Mas como todo esse processo de atualização de repositórios acontece? Bom, todo esse processo se dá a pequenos comandos que irão fazer todo esse processo para a gente.

**sudo apt update** → comando que atualizará a lista de pacotes disponíveis e suas versões, porém não instala e nem atualiza nenhum pacote.

**sudo apt upgrade** → comando que instala as versões mais recentes dos pacotes que você possui instalado na sua máquina.

Quando a **Canonical**, a empresa que desenvolve o Ubuntu, deseja atualizar algum componente do sistema, basta subir uma nova versão do pacote para o repositório, e ele vai aparecer como uma atualização para você.

Para facilitar a organização, geralmente a distros dividem os tipos de pacotes em repositórios diferentes, como por exemplo (no caso do Ubuntu), o repositório “**Multiverse**”, onde você encontrará apenas softwares de código fechado, como drivers de vídeo.

Mas apesar da existência de um servidor como repositório, geralmente é preciso de uma ferramenta que acesse os dados do servidor e baixe e instale os pacotes no seu sistema, para isso existem utilitários como o “**apt**” (sigla para **advanced packaging Tool**), que consegue buscar os pacotes no repositório e trazer para o seu sistema, para que, geralmente outro utilitário o instale, o dpkg.

## Mirrors

O Canonical é uma empresa britânica, e seus servidores principais são hospedados geograficamente muito longe de nós, brasileiros. É aí que entram outros servidores que nos chamamos de **mirrors**.

Os **mirros**, ou espelhos, são servidores que basicamente copiam todo o conteúdo do repositório principal do sistema e estando mais próximos de nós, eles tendem a ter uma velocidade maior, permitindo que o download das aplicações seja mais rápido.

## No mundo Ubuntu temos também os chamados repositórios PPA

**PPA** quer dizer “**Personal Package Archives**” e são originários do serviço de hospedagem e versionamento de software da Canonical, o Launchpad. Lá desenvolvedores podem colocar softwares para o Ubuntu e seus derivados, criando repositórios especiais para as aplicações que não fazem parte dos repositórios padrões do sistema.

## Repositórios offline

Até agora eu falei de repositórios online, que são estes que são hospedados em servidores mundo à fora, mas um repositório de software não precisa ser online. Um exemplo legal disso é o Debian. Se você já tentou baixar alguma vez o sistema, deve ter reparado que existem várias mídias de instalação.

A primeira mídia geralmente inclui o próprio sistema com alguns softwares básicos, porém, nas mídias seguintes você tem a versão offline dos softwares que compõem a distro, sendo um repositório offline de softwares.

O interessante das distros Linux, de forma geral, possuírem repositórios online, é que geralmente são as próprias empresas e sistemas que te oferecem, não somente os softwares que estão instalados na distro em si, mas também os disponíveis para instalar neste repositório de software.

Essa prática aumenta fortemente a segurança do sistema. O GIMP, o LibreOffice e o Firefox que você pode instalar no Ubuntu, para citar alguns exemplos, por mais que sejam feitos por comunidades e empresas terceiras, por serem software livre, são empacotados pela equipe da Canonical e distribuídos dentro do Ubuntu de forma semi-independente.

Por isso o LibreOffice que você pode baixar em formato .deb no site do software não é exatamente igual ao que você encontra no repositório do Ubuntu.

Fonte: **Diolinux.com.br**

## Estrutura de Diretórios

O sistema Linux possui a seguinte estrutura básica de diretórios:

**/bin** Contém arquivos, programas do sistema que são usados com frequência pelos usuários.

**/boot** Contém arquivos necessários para a inicialização do sistema.

**/cdrom** Ponto de montagem da unidade de **CD-ROM**.

**/dev** Contém arquivos usados para acessar dispositivos (periféricos) existentes no computador.

**/etc** Arquivos de configuração de seu computador local.

**/floppy** Ponto de montagem de unidade de disquetes.

**/home** Diretórios contendo os arquivos dos usuários.

**/lib** Bibliotecas compartilhadas pelos programas do sistema e módulos do **kernel**.

**/mnt** Ponto de montagem temporário.

**/usr** Contém maior parte de seus programas. Normalmente acessível somente como leitura.

**/var** Contém maior parte dos arquivos que são gravados com frequência pelos programas do sistema, e-mails, **spool** de impressora, cache, etc.

**/sbin** Diretório de programas usados pelo superusuário (root) para administração e controle do funcionamento do sistema.

## Comandos básicos do Linux

## Listando Arquivos

O comando **ls** mostra o conteúdo de um diretório.

O formato do comando é o seguinte:

**ls [ -l ] [ -a ] [ -F ] [dir]**

Onde **[-l]** é o formato longo, e **[-a]** serve para mostrar todos os arquivos, incluindo arquivos ocultos (os quais têm seu nome indicado por um ponto). Existem várias outras opções, embora estas sejam mais usadas. Finalmente, **[-F]** coloca no final dos nomes de arquivo um símbolo indicando o seu tipo.

Um exemplo do uso do **ls** é mostrado a seguir:

**[mark@kali linux]\$ ls**

LessTif bin doc etc fonts include lib man share

Um exemplo do uso do **ls** usando parâmetros:

**[mark@kali linux]\$ ls -laF**

drwxr-xr-x	11	root	root	4096	Ago 27 2002	./
drwxr-xr-x	22	root	root	4096	Mar 22 2003	../
drwxr-xr-x	3	root	root	4096	Mar 22 2003	LessTif/
drwxr-xr-x	2	root	root	8192	Jul 24 08:45	bin/
drwxr-xr-x	2	root	root	4096	Jul 24 2003	doc/
drwxr-xr-x	4	root	root	4096	Ago 27 2002	etc/
drwxr-xr-x	2	root	root	4096	Mar 22 2003	fonts/
drwxr-xr-x	4	root	root	4096	Mar 22 2003	include/
drwxr-xr-x	9	root	root	4096	Jul 24 2003	lib/
drwxr-xr-x	7	root	root	4096	Jul 24 2003	man/
drwxr-xr-x	15	root	root	4096	Jul 24 08:45	share/

No exemplo acima, como são nomes de diretórios; o parâmetro **[-F]** adiciona uma barra indicando nome de diretório. O parâmetro **[-l]** coloca várias informações sobre o arquivo (permissões, links, dono, grupo, tamanho, data, hora, nome do arquivo).

## Metacaracteres

Existem sinais, chamados metacaracteres, usados para facilitar a utilização de comandos no Linux. Quando se trabalha com os comandos de manipulação de arquivos, freqüentemente é útil empregarmos metacaracteres. Estes símbolos – como **\***, **?**, **[]**, **{}** – são úteis para se referenciar arquivos que possuam características em comum.

Para os exemplos dados nesta seção, será usada a seguinte lista de arquivos:

**[mark@kali linux]\$ ls**

12arquivo 1arquivo 2arquivo arquivo arquivo3 arquivo34 arquivo5arquivo

O asterisco **"\*"**:



O asterisco é usado para representar “qualquer quantidade de qualquer caractere”. Por exemplo, **arquivo\*** retornaria todos os arquivos em que o nome iniciasse com “arquivo”. Veja o efeito da utilização prática deste **meta caractere**.

```
[mark@kali linux]$ ls arquivo*
```

```
arquivo arquivo3 arquivo34 arquivo5arquivo
```

O Ponto de Interrogação “?”:

O ponto de interrogação é utilizado para representar “um único e qualquer caractere”. Ao digitar **ls arquivo?**, o usuário estará pedindo a lista de arquivos cujos nomes são indicados por “arquivo” e terminal com um único caractere qualquer. Como no exemplo que segue:

```
[mark@kali linux]$ ls arquivo?
```

```
Arquivo3
```

Agora digitando **ls ??arquivo** o resultado seria:

```
12arquivo
```

Os colchetes “[ ]”:

Os colchetes são utilizados para indicar uma lista de caracteres. Para entender melhor; verifique os exemplos.

```
[mark@kali linux]$ arquivo[123] arquivo3
```

```
[mark@kali linux]$ [123]arquivo 1arquivo 2arquivo
```

As chaves “{ }”:

As chaves têm sua utilização muito assemelhada a dos colchetes. A diferença está na possibilidade de referenciar **seqüências** de caracteres separadas por vírgulas, conforme o exemplo a seguir:

```
[mark@kali linux]$ ls arquivo{1,34} arquivo34
```

```
[mark@kali linux]$ ls arquivo{1,2,3,34} arquivo3 arquivo34
```

## Criando e Removendo Diretórios

### 1 - Criando Diretórios

O comando **mkdir** é usado para criar diretórios. A sintaxe do comando será mostrado a seguir:

```
mkdir [parâmetros] nome_dir
```

A linha de comando a seguir cria um diretório:

```
[mark@kali linux]$ mkdir meu_diretorio
```

O comando também pode ser usado para criar uma árvore de diretórios, como será mostrado a seguir:

```
[mark@kali linux]$ mkdir -p meu_dir/meu_sub_dir/sub_sub_dir
```

O comando anterior cria um diretório chamado meu\_dir, e dentro dele cria um subdiretório chamado meu\_sub\_dir e dentro deste, um subdiretório chamado sub\_sub\_dir.

Também é possível criar vários diretórios em simultâneo; simplesmente colocando vários nomes de diretórios junto com o comando, como será mostrado a seguir:

```
[mark@kali linux]$ mkdir dir_1 dir_2 dir_3
```

O comando anterior criará os diretórios dir\_1, dir\_2 e dir\_3 dentro do diretório atual.

## Removendo Diretórios

O comando rmdir é usado para remover diretórios. Por exemplo; para remover o diretório meu\_dir basta digitar o seguinte comando:

```
[mark@kali linux]$ rmdir meu_dir
```

O comando também pode remover árvores de diretórios. Para tal, utiliza-se o parâmetro [-p ], como será mostrado a seguir:

```
[mark@kali linux]$ rmdir -p temp/sub_dir/sub_dir_2
```

O comando anterior apagou o subdiretório sub\_dir\_2, depois apagou o subdiretório sub\_dir e finalmente apagou o temp.

Porém, o comando remove diretórios e não arquivos; se existir algum arquivo dentro do diretório este não será removido. Para conseguir remover diretórios com arquivos deve-se utilizar em conjunto o comando **rm**, que será visto mais adiante.

## Copiando Arquivos

O comando cp é utilizado para efetuar a cópia de arquivos no Linux, Sua sintaxe é mostrada a seguir:

```
cp [parametros] arquivo_original [destino]
```

Observações importantes relativas à cópia de arquivos:

- \* Copiar um arquivo para outro diretório onde já existe outro arquivo com mesmo nome: o arquivo será sobrescrito.
- \* Copiar um arquivo para outro diretório que, por sua vez, possui um diretório com mesmo nome do arquivo a ser copiado: não é permitido, pois no Linux um diretório também é um arquivo.
- \* Copiar um arquivo, especificando como arquivo\_destino outro nome: o arquivo será renomeado durante a cópia.

<b>Opção</b>	<b>Descrição</b>
-a	Preserva as permissões do arquivo_original quando possível.
-b	Faz backup de arquivos que serão sobrescritos.
-i	Solicita confirmação antes de sobrescrever arquivos.
-R	Copia diretórios recursivamente, ou seja, toda a árvore abaixo do diretório de origem. O destino sempre será um diretório.

Iniciando pela forma mais simples do comando, ou seja, copiar um arquivo para um novo arquivo. O comando pode ser visto a seguir:

```
[mark@kali linux]$ cp doc.txt documento.txt
```

Neste caso ocorre a criação do arquivo **documento.txt** a partir do arquivo **doc.txt**. Também é possível copiar para outro local como será mostrado a seguir:

```
[mark@kali linux]$ cp doc.txt /tmp
```

Como não foi mencionado o nome do arquivo de destino, será criado um arquivo com o mesmo nome do atual, É sempre bom ter um pouco de cuidado no uso do comando **cp**; para tal será usado o parâmetro -i. A menos que se utilize essa opção, o comando **cp** irá sobrescrever os arquivos existentes, como pode ser visto a seguir:

```
[mark@kali linux]$ cp -i doc.txt /tmp  
cp: sobrescrever `/tmp/doc.txt'? y
```

Um método mais seguro seria usar o parâmetro -b (backup); então, quando o cp encontra um arquivo com o mesmo nome cria uma cópia acrescentando um “~” ao nome do arquivo. Como pode-se observar a seguir:

```
[mark@kali linux]$ cp -b doc.txt /tmp  
[mark@kali linux]$ ls /tmp/doc.txt*  
doc.txt doc.txt~
```

Também é possível copiar vários arquivos simultaneamente; para tal, basta colocar os nomes dos arquivos a copiar logo depois do comando, como mostrado a seguir:

```
[mark@kali linux]$ cp arq_1 arq_2 arq_3 arq_4 dir_1
```

Vale lembrar que o último nome na cadeia é o destino, ou seja, os arquivos **arq\_1**, **arq\_2**, **arq\_3** e **arq\_4** são copiados para o diretório **dir\_1**.

## **Mover ou Renomear**

As habilidades para mover e renomear arquivos no Linux são básicas para organizar informações no sistema. A seguir, será apresentado como fazê-lo utilizando o comando **mv**.

O formato básico do comando é mostrado a seguir:

```
mv arquivo destino
```

O comando mv é basicamente usado pra mover um arquivo dentro do sistema de arquivos do Linux.

```
[mark@kali linux]$ mv documento.txt /tmp
```

O comando anterior move o arquivo documento.txt para o diretório /tmp. É possível também usar o comando para renomear arquivos, como mostrado no exemplo a seguir:

```
[mark@kali linux]$ mv doc.txt documento.txt
```

Para não sobrescrever arquivos deve-se utilizar o parâmetro -i, como mostrado a seguir:

```
[mark@kali linux]$ mv -i doc.txt documento.txt  
mv: sobrescrever `documento.txt`? Y
```

```
[mark@kali linux]$ ls documento*  
documento.txt documento.txt~
```

Também é possível renomear um arquivo durante a movimentação do mesmo:

```
[mark@kali linux]$ mv documento.txt /tmp/documento-2.txt
```

No exemplo a seguir será movido um diretório:

```
[mark@kali linux]$ mv dir_1 dir_2
```

O comando acima move toda a árvore do dir\_1 para dentro do dir\_2. Caso o dir\_2 não exista, o dir\_1 será renomeado para dir\_2.

## Removendo Arquivos

O comando rm (remove) é usado para remover arquivos e diretórios. É possível remover vários arquivos simultaneamente, bastando para tal colocar o nome dos arquivos a remover, logo depois do comando. O formato básico do comando é mostrado a seguir:

```
rm [parametros] arquivo
```

Como primeiro exemplo será emitido um comando para apagar o arquivo documento.txt.

```
[mark@kali linux]$ rm documento.txt
```

É também possível remover vários arquivos listados logo após o comando. Por exemplo:

```
[mark@kali linux]$ rm documento.txt doc.txt documento-2.txt
```

A maneira mais segura de se usar o comando rm é com o parâmetro -i, ou seja, é solicitada uma confirmação para cada arquivo a apagar.

```
[mark@kali linux]$ rm -ri /tmp
```

No comando anterior, além de se usar o parâmetro **-i** foi também usado o parâmetro **-r** (recursivo), isto remove todos os arquivos do diretório **/tmp** de forma recursiva. Já o parâmetro **-i** irá pedir uma confirmação para cada arquivo a apagar.

Caso não seja necessária uma confirmação, pode-se forçar a remoção de toda a árvore de diretórios; para tal utiliza-se o parâmetro **-f**.

```
[mark@kali linux]$ rm -rf /tmp
```

O que ocorre é a remoção total do diretório e de todos os seus subdiretórios.

Lembre-se de evitar o uso do comando **rm** desnecessariamente quando estiver trabalhando como **root**, ou seja, superusuário, para prevenir que arquivos necessários ao sistema sejam apagados acidentalmente.

É possível utilizar o comando com meta caracteres, conforme mostrado a seguir:

```
[mark@kali linux]$ rm *.txt *.doc
```

## Fim dos Estudos

Se você chegou até aqui, lembre-se sempre de uma coisa!

“Jamais, nunca nessa vida desista! Persista, continue, mas um pouco e... tudo dará certo.”

**Fiquem bem, espero vocês no meu próximo livro!**

**Abraços do Mark e até breve, gente :)**