

Escrito por: Artur de Oliveira Barbosa  
GitHub: <https://github.com/Space2544>  
Email: [arturoliveira886@gmail.com](mailto:arturoliveira886@gmail.com)

“conhecimento é poder” -Francis Bacon

---

## Documentação Shell & Shell Script

Shell do inglês “casca” nesse caso de um sistema operacional, é onde a maioria dos processos acontecem, e claro você pode usar isso ao seu favor para por exemplo criar um arquivo que pode ser lido no “shell” que é apelidado de “Script”, no caso “Shell Script”, um script pode funcionar de inúmeras maneiras inclusive para automação de processos no linux ou no windows por exemplo.

E claro esse processo requer um conhecimento sobre o sistema, que no nosso caso é o linux ou melhor dizendo, Unix, já que o linux é da família unix, e no windows o processo é semelhante em alguns pontos, como o comando *echo* que serve pra mostrar algo na tela.

Lembrando que normalmente se caso você precisar de ajuda em relação o que mais cada comando faz você pode colocar `--help` na frente de determinada função.

Exemplo: **`mkdir --help`**

---

## Comandos básicos

O shell do linux tem uma curva de aprendizado bem razoável e claro, só de aprender shell scripting você facilita e muito sua vida usando o linux e demais distribuições ou até mesmo alguns sistemas da família Unix.

Por exemplo você pode pegar um comando como: `sudo apt update && sudo apt upgrade` e resumir em uma palavra só, como por exemplo `update`.

Mas antes de aprender isso, vamos ao basico.

Comandos:

**cd**: **c**(hange) **d**(irectory), esse comando serve para trocar de diretorio, que por padrão você está na `/home`, mas quer se deslocar para outro diretorio, então o comando **cd** é necessario.

Vamos supor que você queira entrar no diretorio `downloads`, mas você não sabe como fazer isso no terminal pra instalar algo ou que alguma coisa precisa ser feita dentro dessa pasta, pra isso você vai dar o comando:

(bash) \$ cd Downloads/ (obs, não precisa do \$, isso é pra representar o bash).

Mas se caso esse não for o nome da pasta você pode apertar a tecla Tab pra mostrar quais os possíveis diretórios que podem ser acessados dentro da /home por exemplo:

\$ cd (pressionou tab)

Downloads/	Música/	.config/
Documentos/	Modelos/	.local/

Esse exemplo acima foi o que pode aparecer no terminal quando você digita **cd**, e nesse exemplo mostra também os diretórios que começam com ponto.

Esses diretórios ou arquivos são “escondidos” já que são do sistema ou que é pra ser escondido mesmo, por exemplo se eu quiser criar um arquivo escondido eu posso usar **.(nome do arquivo ou pasta)** e assim ficará “invisível” quando usar um gerenciador de arquivos.

Mas por exemplo se caso você quiser modificar algo, normalmente fica dentro da pasta **.config**, e nisso você tem um leque de possibilidades muito grande, e daí acaba surgindo customizações na interface da sua distribuição linux.

Bem mas como faria pra retornar se caso eu entrei na pasta downloads?, pra isso você precisa adicionar dois pontos depois do **cd**.

Exemplo:

/Downloads \$ **cd ..** (pressionou enter.)

\$ (voltou para o diretório de antes.)

e se você quiser retornar de vez sem ser um diretório por vez, apenas digite **cd** e vai retornar para a /home.

=====

**ls**: list, **ls** é um comando que serve para listar arquivos e pastas ou diretórios, ou seja você pode ter uma visualização fixa de quais diretórios e arquivos estão dentro de por exemplo /downloads ou /home, e só é necessário dar apenas **ls** para mostrar o conteúdo dentro.

Exemplo:

/Download \$ **ls**

image001.jpg

image002.jpg

image003.jpg  
image004.jpg  
image01.svg  
image1.png

=====

**mkdir**: **ma**ke **dir**ectory, como o próprio comando já diz, faz um diretório, e claro, pra que esse diretório possa ser criado é necessário fazer o seguinte, vamos usar o Downloads como uma pasta para esse exemplo:

```
$ sudo mkdir Downloads/teste      (teste vai ser a pasta a ser criada).
```

Mas você se perguntou o que é sudo?, sudo é uma maneira de fazer com que você através desse comando tenha “poder” administrativo ou seja você “vira” um super usuário ao dar esse comando, normalmente não é necessário esse comando mas se caso aparecer um aviso de que “você precisa ser um super usuário para fazer tal ação”, aí você pode utilizar, mas toma muito cuidado, qualquer ação que altere algo do sistema e que o sistema avisa que pode haver alterações, primeiro pesquisa e tente entender o que você está fazendo e como que isso pode afetar o sistema, dado o aviso vamos voltar ao assunto.

Depois de ter criado a pasta teste em Downloads/, você pode criar arquivos com o comando **touch**, touch serve para “tocar” criar um arquivo com a extensão (.txt, .svg, .jpg, .docx ...) de sua preferência e necessidade, logo temos algo que facilita e muito quando quisermos criar um arquivo dentro de alguma pasta sem precisar necessariamente de uma interface guiada, (conhecida também por GUI) logo para se criar um arquivo é usado o comando:

```
$ touch 'nome do arquivo. extensão de sua preferência'
```

e assim você cria um arquivo com **touch**, mas vamos criar um arquivo que seja adequado:

```
Downloads/teste $ touch 'teste_001.txt'
```

Quando adicionamos \_ o texto fica armazenado sem problemas e é mais recomendável assim visto que pode causar problemas no arquivo e dentre outras coisas. Agora você tem uma pasta, um arquivo, conhecimento de como sair e entrar dentro de pastas, mas como fazer pra editar esse arquivo de texto?

Uma das várias opções que temos é o editor de texto, mas qual?, por padrão como o Linux é kernel do Gnu, logo temos inúmeras ferramentas que aliás estão disponíveis na Free Software Foundation, se caso quiser se aprofundar mais por lá é só entrar no link: <https://www.fsf.org/pt-br> (é recomendável que se pesquise o nome em si da FSF por segurança.)

E dentro do Gnu/Linux temos um editor de texto que já vem por padrão que se chama **nano**, nano é um editor de texto bem simples para editar/escrever/modificar textos, assim fazendo com que seja bem fácil a sua utilização. E para chamar esse editor de texto é bem simples:

Downloads/teste \$ nano teste\_001.txt

(pode apertar tab pra selecionar/auto-completar o nome do arquivo sem precisar digitar tudo).

E então você estará na tela do