

Comandos comuns do Linux — Uma introdução

Eduardo Ramos

Introdução

O objetivo desse documento é ser uma lista de comandos para se usar no terminal, que você pode acessar nas distros mais comuns por `ctrl + alt + T`. Sugiro que comecem por uma distro que seja baseada no Arch, porque é possível obter acesso ao repositório AUR (Arch User Repository), que dá acesso a uma quantidade gigantesca de programas.

Se você for seguir esse conselho, as melhores para começar provavelmente são ou [Manjaro](#) ou [Arcolinux](#). Se for usar o Arcolinux, sugiro especificamente o ArcolinuxL para começar. No site do Arcolinux, o criador e desenvolvedor Erik Dubois colocou uma sugestão de [caminhos a se seguir](#) conforme for aprendendo a usar Linux através dessa distro.

Um bom jeito de começar é através de máquinas virtuais, mas eu comecei por dualboot, que permite poder escolher entre um sistema operacional ou outro assim que eu ligava o computador. Encontrei [este vídeo no Youtube](#) que é bem parecido com o processo que eu fiz ao instalar o Manjaro pela primeira vez, fazendo dualboot com o Windows 10. A única diferença foi que não precisei configurar o dualboot: colocando prioridade maior para o Manjaro, e não para o Windows 10, o próprio Manjaro me dava opção ou de bootar ele mesmo ou o Windows. Hoje em dia não uso nem mais o Manjaro nem o Windows 10, então não posso garantir que está parecido. Para o Arcolinux, o próprio desenvolvedor posta frequentemente vídeos de como instalar cada uma das opções que sua distro dá, como [esta falando do ArcolinuxL](#), por exemplo.

Talvez eu faça um vídeo futuramente ensinando como instalar esses sistemas operacionais, mas basicamente é só instalar no pendrive o ISO que está nos links que coloquei acima, do Manjaro ou do Arcolinux, usar um software como o Balena Etcher para instalar esse ISO no pendrive, deixar ele conectado ao computador e, com ele conectado, reiniciar a máquina e entrar na bios (geralmente por F2 F9 F12 ou Del) para colocar o pendrive com o ISO com prioridade para ser lido antes do Windows.

Conceitos e analogias com Windows

Vários conceitos de Linux são análogos a conceitos de Windows.

Conceitos

- Directory/diretório \longleftrightarrow pasta/folder;
- Directory tree \longrightarrow “caminho” do arquivo, com diretórios separados por /. Exemplos:
 - Windows: C:\Users\user\...;
 - Linux: /home/user/...;
- Caminho absoluto/absolute path \longrightarrow localização de um arquivo. Exemplos:
 - /home/johndoe/Documents/phone-list.txt;
 - /home/johndoe/Music/Bob-Dylan/Highwat-61-Revisited/Desolation-Row.mp3;

Tipos de diretório

- Root directory \longrightarrow a pasta mais acima na árvore de pastas, todas de alguma forma são subdiretórios dela \longrightarrow **representação:** /;
- Home directory \longrightarrow database do usuário \longrightarrow **representação:** ~;
- bin (binary) \longrightarrow programas executáveis;
- opt (optional) \longrightarrow software comercial;

Tipos de comando

- Programas executáveis;
- Arquivos “built-in” de shell;
- Scripts de shell;
- Alias: comandos criados;

Links e inode

Index node (inode) é informação sobre o arquivo exceto nome e conteúdo (todo arquivo só tem um index node específico).

- Soft link \longrightarrow atalho, tem inode diferente;
- Hard link \longrightarrow cópia, tem inode igual, e atualiza sempre de acordo com o original;

Linha de comando

Agora de fato para a linha de comando no terminal.

Navegação e comandos básicos

- `ctrl + alt + T` → Abre o terminal;

A partir daqui, basicamente só teremos comandos:

- **pwd** → “print working directory”: mostra o caminho absoluto do diretório onde o terminal está no momento;
- **ls DIRECTORY** → lista conteúdo dentro do diretório `DIRECTORY` e, se não especificar o diretório, lista o conteúdo do diretório onde o terminal está
 - `ls -i` → lista os inodes dos arquivos;
 - `ls -l` → lista o tamanho dos arquivos em bytes;
 - `ls -t` → lista a data de modificação dos arquivos;
 - `ls -r` → lista os arquivos em ordem reversa;
 - `ls -R` → lista os conteúdos do diretório e subdiretórios;
 - `ls -a` → lista todos os arquivos, incluindo arquivos escondidos;
 - * Arquivos escondidos começam com `.` em seus nomes. Exemplo: `~/.bashrc` é um arquivo escondido no diretório home, `~`;
- **cd** → troca o diretório onde você está na janela do terminal;
- **type** → mostra o tipo de comando;
- **file** → descreve o arquivo (extensão, tamanho, etc.);
- **touch** → atualiza a timestamp do arquivo e, se usar em um arquivo que não existe, cria um arquivo com esse nome;
 - `touch /path/to/file` → usa `touch` em um arquivo específico num caminho específico;
- **mkdir NOME** → cria um diretório do zero chamado `NOME`;
- **rm NOME** → remove arquivo chamado `NOME`, desde que não seja um diretório;
 - `rm -i NOME` → pede confirmação extra para remover;
 - `rm -R NOME` → remove `NOME` se ele for um diretório, pedindo confirmação;
 - `rm -f NOME` → remove `NOME` de maneira forçada, sem pedir confirmação (serve para diretórios);
 - `rm -v NOME` → faz um sumário do que foi removido (se for um diretório, mostra tudo);
- **mv NOME1 NOME2** → move `NOME1` para um diretório `NOME2` ou, se `NOME2` não for um diretório que existe, renomeia `NOME1` para `NOME2`;
 - É útil para fazer arquivos escondidos ficarem visíveis;

- `ln NOME1 NOME2` → cria um hard link, não funciona para diretórios;
 - `ln -s NOME1 NOME2` → cria um soft link;
- `cp NOME1 ... NOME2 DESTINO` → copia arquivos NOME1 ... NOME2 para DESTINO;
 - `cp -R` → funciona como `cp` mas para diretórios;
 - `cp -i` → funciona como `cp` mas pede por confirmação;
 - `cp -r` → funciona como `cp` mostra um resumo do que foi copiado;
- `sudo` → dá autorização de administrador para programas;
- **history** → mostra o histórico de comandos utilizados no terminal;
 - **history -c** → limpa o histórico de comandos usados;

Importantes

Na linha de comando, alguns caracteres são importantes de se conhecer.

- `comando1 && comando2` → realiza os dois comandos em questão em sequência, primeiro um e depois outro;
- `*` → representa uma combinação qualquer de caracteres;
- `?` → representa um caracter qualquer. Exemplos de aplicação de `*` e `?`:
 - `ls *` → lista todos os arquivos do diretório atual, essencialmente igual a `ls`;
 - `ls a*` → lista todos os arquivos do diretório atual que começam com o caracter `a` e têm absolutamente qualquer combinação de caracteres depois;
 - `ls ?.mp3` → lista todos os arquivos do diretório atual que têm no nome apenas um caracter seguido da extensão `.mp3`;
 - `ls *.mp?` → lista todos os arquivos do diretório atual que são ou `.mpa`, `.mpb`, ..., `.mp1`, `.mp2`, `.mp3`, `.mp4`, ..., ou qualquer outro símbolo no lugar de `?`.

Comandos de texto com nano e less

Existem vários arquivos que trabalham com texto que vêm com a instalação do Manjaro ou Arcolinux, e a maioria das distros. Vou focar em um, `nano`, mas existem outros comandos para ler arquivos como `less`, `cat`, `tac`, `head` e `tail`.

Além disso, o comando `wc` (abreviando word counter) mostra o número de linhas, palavras e bytes, que é o mesmo número de caracteres.

`nano` é um comando bem simples. Um editor de texto similar ao bloco de notas, de nível bem introdutório. Nele, temos

- `tab` → autocomplete;
- `ctrl+A` → vai para o primeiro caracter de uma linha;

- `ctrl+D` → deleta o caracter selecionado;
- `ctrl+E` → vai para o fim da linha;
- `alt+F` → vai uma palavra para a frente;
- `alt+B` → vai uma palavra para trás;
- `alt+U` → deixa a letra selecionada maiúscula;
- `alt+L` → deixa a letra selecionada minúscula;
- `ctrl+K` → corta do caracter selecionado até o final da linha;
- `ctrl+U` → corta do caracter selecionado até o começo da linha;
- `ctrl+L` → clear.

Além disso, tem as mais simples na parte inferior de quando abrimos o programa. Basta substituir `^` por `ctrl +`.

O `less` é ainda mais simples. Temos

- `q` → para sair do programa;
- `g` → para ir para o começo do arquivo;
- `G` → para ir ao fim do arquivo;
- `/texto` → para procurar por “texto”;

Atualização, instalação de programas e alias

Para instalar programas novos no Manjaro, no Arcolinux e outras distros baseadas em Arch, o comando a ser usado é `pacman`. Em outros como Debian, Ubuntu ou Mint é `apt-get`. Como estou focando em distros baseadas em Arch, vou seguir com o primeiro.

Para atualizar o sistema, temos

```
# pacman -Syu
```

Pode ser usado `pacman -Syu`, mas com dois `y` tem menos chance de dar problemas.

Para instalar programas e pacotes novos,

```
# pacman -S package1-name ... packageN-name
```

Onde, obviamente, se substitui `package-name` pelo nome do programa que você quer, e pode ser instalado mais de um simultaneamente. O primeiro a ser baixado é o `yay`, uma espécie de ajudador de instalação que te dá acesso ao repositório de usuários de Arch Linux, permitindo encontrar mais programas que pelo `pacman` em si seria mais complicado.

Provavelmente o pacote `pacman` não vai rodar sem dar permissão de administrador para ele, então é necessário usar, na verdade, `sudo pacman -S` e derivados.

Mencionei antes o `yay` como um ajudador de instalação. Para usá-lo, é exatamente com os mesmos complementos do `pacman`, isto é:

```
# yay -S package1-name ... packageN-name
```

É um comando para procurar no AUR os pacotes `package1-name` até `packageN-name` e instalar os que forem encontrados.

Para desinstalar programas, temos um comando simples envolvendo o `pacman`:

```
# pacman -R package1-name ... packageN-name
```

Assim, os pacotes `package1-name ... packageN-name` são desinstalados.

É importante destacar que o `yay` também pode ser usado para atualizar o kernel e desinstalar programas, mas não é sugerido que se faça isso.

Alias

Alias é como se chama a maneira de se criar comandos no Linux. Caso você não queira usar sempre `sudo pacman -Syyu`, pode criar um comando para isso, por exemplo.

Para fazer isso, vá para o diretório `home`, `~` e edite um arquivo oculto chamado `.bashrc` com o `nano`. Para isso, rode no terminal:

```
# cd ~ && nano .bashrc
```

Dentro do arquivo, basta criar uma linha, preferencialmente com um comentário antes para ficar organizado (comentários são linhas que começam com `#`) para explicar o que cada comando faz, com algo nas linhas de¹

```
alias nome-para-o-comando='codigo-do-comando'
```

Então, se você quiser criar um comando para atualizar o sistema que não seja tão longo quanto com o `pacman`, basta adicionar uma linha como:

```
alias update='sudo pacman -Syyu'
```

E, ao reiniciar o computador, esse comando estará valendo. Na próxima subseção dou outro exemplo útil de alias.

Programas úteis

Aqui alguns programas úteis, mais leves que consigo usar de maneira funcional, open-source que eu sugiro.

- **mpv**: tocador de vídeos e áudios. Provavelmente o Manjaro vem com VLC instalado, mas `mpv` é melhor e bem mais rápido;
- **Brave**: navegador;
- **zathura**: leitor de pdf;

¹Os códigos envolvendo alias vão ficar meio estranhos porque por algum motivo quando eu tentei fazer rodar isso no \LaTeX ele estava dando problema por causa das aspas que são **necessárias** para o alias funcionar. Depois se eu encontrar solução atualizo o documento.

- Se quiser ter esse documento em mãos para lembrar de mais comandos facilmente e tiver ele salvo no diretório/pasta Downloads, basta adicionar ao arquivo `.bashrc` a seguinte linha:

```
alias ajuda="zathura ~/Downloads/commands.pdf"
```

E então agora se usar o comando `ajuda` na linha de comando do terminal, esse documento será aberto.

- **tar, unzip, rar**: três programas para lidar com arquivos compactados, respectivamente `.tar`, `.zip` e `.rar`;
- **deluge**: gerenciador de torrents;
- **ffmpeg**: talvez demore um pouco para usar ele por linha de comando em vez de um OBS, mas é mais útil para gravar vídeos e áudios simples. O OBS ainda é melhor para fazer stream;
- **pulsemixer**: gerenciador de áudio;
- **htop**: gerenciador de tarefas;
- **newsboat**: leitor de RSS;
- **vim**: editor de texto superior ao nano e, basicamente, a qualquer outro, mas exige certo trabalho para se acostumar, então não comece por aqui a não ser que esteja disposto a gastar um certo tempo.

Como se virar sozinho

Usar Linux vai te ensinar a pesquisar. Use ferramentas como SearX, DuckDuckGo, ou mesmo Google para entrar em fóruns. O mais comum de ter respostas é provavelmente o [Stacks Overflow](#), mas fóruns do Arch para quem escolher Manjaro ou Arcolinux serão úteis também.

Existe um manual para cada programa que é acessível pelo terminal. O comando é basicamente

```
# man package-name
```

Novamente, só substituir `package-name` pelo programa que você precisa aprender a usar, e então uma aba do less vai abrir no terminal com o manual do software.

Contato

Vou tentar atualizar isso aqui se eu lembrar de algo que esqueci. Pode tirar dúvidas também.

- edramoscm@protonmail.com — Mande perguntas!
- eduardoramos.xyz — Quase tudo que quero deixar público tá aí!
- [RSS](#) — RSS feed para acompanhar o que eu faço!
- Falando em RSS, [Odysee](#) — Canal do Odysee onde posto vídeos, inclusive um sobre como usar RSS no Linux (na época, gravado quando eu usava Manjaro).