

Contents

Guia Completo de Atalhos e Comandos para Terminais (Bash, PowerShell e CMD)	2
Índice	2
1. Atalhos Comuns de Terminal (Bash, PowerShell e CMD)	2
2. Atalhos Específicos do Terminal Bash	3
2.1. Edição e Controle de Linha:	4
2.2. Histórico e Execução de Comandos:	4
2.3. Controle de Processos:	4
2.4. Gerenciamento da Tela:	5
3. Atalhos Específicos do PowerShell	5
3.1. Edição de Texto e Comando	5
3.2. Controle de Execução	5
3.3. Histórico de Comandos	5
3.4. Interface do PowerShell ISE (Integrated Scripting Environment)	6
4. Comandos Comuns e Específicos para Bash, CMD e PowerShell	6
5. Comandos Comuns (funcionalidade similar ou equivalentes)	6
5.1. Navegação de Diretórios:	6
5.2. Listar Conteúdo do Diretório	7
5.3. Criar Diretório	7
5.4. Remover Arquivo	7
5.5. Remover Diretório (Vazio)	7
5.6. Remover Diretório (Não Vazio/Recursivo)	8
5.7. Copiar Arquivos	8
5.8. Mover/Renomear Arquivos	8
5.9. Exibir Conteúdo de Arquivo	9
5.10. Limpar Tela	9
5.11. Testar Conectividade de Rede	9
5.12. Configuração de IP (Exibir)	9
5.13. Sair do Terminal	9
6. Comandos Específicos do Bash	9
7. Comandos Específicos do CMD (Prompt de Comando do Windows)	10
8. Comandos Específicos do PowerShell	11
9. Gerenciamento de Variáveis de Ambiente	12

Guia Completo de Atalhos e Comandos para Terminais (Bash, PowerShell e CMD)

Este guia detalhado explora os atalhos de teclado e comandos essenciais para otimizar sua produtividade em diferentes ambientes de linha de comando: Bash (comum em sistemas Linux e macOS), PowerShell (o shell de script e automação da Microsoft) e CMD (o Prompt de Comando tradicional do Windows). Compreender e dominar essas ferramentas é fundamental para qualquer desenvolvedor, administrador de sistemas ou entusiasta que busca eficiência na interação com o computador.

Índice

1. Atalhos Comuns de Terminal (Bash, PowerShell e CMD)
2. Atalhos Específicos do Terminal Bash 2.1. Edição e Controle de Linha 2.2. Histórico e Execução de Comandos 2.3. Controle de Processos 2.4. Gerenciamento da Tela
3. Atalhos Específicos do PowerShell 3.1. Edição de Texto e Comando 3.2. Controle de Execução 3.3. Histórico de Comandos 3.4. Interface do PowerShell ISE (Integrated Scripting Environment)
4. Comandos Comuns e Específicos para Bash, CMD e PowerShell
5. Comandos Comuns (funcionalidade similar ou equivalentes) 5.1. Navegação de Diretórios 5.2. Listar Conteúdo do Diretório 5.3. Criar Diretório 5.4. Remover Arquivo 5.5. Remover Diretório (Vazio) 5.6. Remover Diretório (Não Vazio/Recursivo) 5.7. Copiar Arquivos 5.8. Mover/Renomear Arquivos 5.9. Exibir Conteúdo de Arquivo 5.10. Limpar Tela 5.11. Testar Conectividade de Rede 5.12. Configuração de IP (Exibir) 5.13. Sair do Terminal
6. Comandos Específicos do Bash
7. Comandos Específicos do CMD (Prompt de Comando do Windows)
8. Comandos Específicos do PowerShell
9. Gerenciamento de Variáveis de Ambiente
10. Aliases: Criando Atalhos Personalizados para Comandos

1. Atalhos Comuns de Terminal (Bash, PowerShell e CMD)

Estes atalhos são amplamente utilizados e funcionam de forma semelhante em muitos terminais modernos, embora o comportamento exato possa variar ligeiramente dependendo do emulador de terminal (como Windows Terminal, GNOME Terminal, iTerm2) e do shell.

- **Ctrl + C**: Interrompe o comando ou processo atual em execução. Este é um dos atalhos

mais universais e importantes para parar processos que estão travados ou demorando muito.

- **Ctrl + A:**

- **Bash:** Move o cursor para o **início da linha**.
- **PowerShell (Console):** Em algumas configurações ou emuladores de terminal como o Windows Terminal, pode selecionar todo o texto na linha de comando. No entanto, para mover o cursor para o início da linha, o mais comum é usar a tecla Home ou Ctrl + Home.

- **Ctrl + E:** Move o cursor para o **final da linha**.

- **Ctrl + V:** Cola o texto copiado ou recortado. Este atalho é amplamente suportado na maioria dos terminais modernos.
- **Tab:** Auto-completa comandos, nomes de arquivos e diretórios. Pressione repetidamente para ciclar pelas opções disponíveis. Essencial para agilizar a digitação e evitar erros.
- **Ctrl + L:** Limpa a tela do terminal, mantendo a linha de comando atual no topo. Funciona em Bash e PowerShell.
- **Ctrl + D:** Envia um sinal de “fim de arquivo” (EOF). Pode ser usado para encerrar a entrada de dados (por exemplo, ao digitar texto para um comando como cat) ou para sair da sessão do terminal (similar a exit) se a linha de comando estiver vazia.
- **Ctrl + Shift + T:** Abre uma nova aba no terminal. Este atalho é comum em muitos emuladores de terminal modernos (como GNOME Terminal, Windows Terminal).
- **Ctrl + Shift + N:** Abre uma nova janela no terminal. Comum em muitos emuladores de terminal.
- **Ctrl + Shift + W:** Fecha a aba/janela atual no terminal. Comum em muitos emuladores de terminal.

Observações sobre atalhos de edição: * **Ctrl + X:** No contexto de terminais de console, Ctrl + X **não** é um atalho universal para “recortar” texto como em editores gráficos. Em Bash, Ctrl + X seguido de Ctrl + E pode abrir um editor externo para a linha de comando. Em PowerShell ISE, pode recortar texto selecionado. * **Ctrl + Z:** Em ambientes de console (Bash e PowerShell), Ctrl + Z geralmente **pausa** o processo atual, colocando-o em segundo plano. Não é um “desfazer” de edição de texto. Em PowerShell ISE, Ctrl + Z funciona como “desfazer” para edição de texto.

2. Atalhos Específicos do Terminal Bash

O Bash (Bourne Again SHell) é o shell padrão na maioria das distribuições Linux e macOS. Estes atalhos são otimizados para navegação e edição de linha de comando, aproveitando as capacidades do Readline, a biblioteca de edição de linha do Bash.

2.1. Edição e Controle de Linha:

- **Ctrl + B:** Move o cursor um caractere para trás.
- **Alt + B:** Move o cursor uma palavra para trás.
- **Ctrl + F:** Move o cursor um caractere para frente.
- **Alt + F:** Move o cursor uma palavra para frente.
- **Ctrl + W:** Apaga a palavra anterior ao cursor. O texto apagado é armazenado em um “kill ring” (buffer de recorte).
- **Ctrl + U:** Apaga tudo do cursor até o início da linha. O texto apagado é armazenado no “kill ring”.
- **Ctrl + K:** Apaga tudo do cursor até o final da linha. O texto apagado é armazenado no “kill ring”.
- **Ctrl + Y:** Cola o texto que foi apagado com Ctrl + U, Ctrl + K ou Ctrl + W (o conteúdo do “kill ring”).
- **Alt + D:** Apaga a palavra à frente do cursor.
- **Ctrl + _:** Desfaz a última alteração na linha de comando.

Observação: O atalho Ctrl + Alt + T é um atalho de ambiente de desktop (como GNOME ou KDE) para abrir o terminal, e não um atalho interno do Bash.

2.2. Histórico e Execução de Comandos:

- **Ctrl + R:** Pesquisa incremental inversa no histórico de comandos. Comece a digitar e o Bash mostrará o comando mais recente que corresponde. Pressione Ctrl + R novamente para navegar entre os resultados mais antigos.
- **Ctrl + G:** Sai do modo de pesquisa do histórico (Ctrl + R).
- **Ctrl + P ou Seta para Cima (↑):** Exibe o comando anterior no histórico.
- **Ctrl + N ou Seta para Baixo (↓):** Exibe o próximo comando no histórico.
- **!!:** Repete o último comando executado. Útil para reexecutar rapidamente.
- **!\$ ou Alt + .:** Insere o último argumento do comando anterior. Extremamente útil para reutilizar caminhos de arquivo ou nomes de diretório.
- **!string:** Repete o comando mais recente que começa com string. Ex: !apt para repetir o último comando apt.
- **!string:p:** Imprime o comando que !string executaria, sem executá-lo.

2.3. Controle de Processos:

- **Ctrl + Z:** Pausa o comando atual (coloca-o em segundo plano). Use fg para trazê-lo de volta para o primeiro plano ou bg para continuar em segundo plano.

- **Ctrl + S**: Pausa a saída do terminal. Útil quando um comando está gerando muita saída rapidamente.
- **Ctrl + Q**: Retoma a saída do terminal após uma pausa com Ctrl + S.

2.4. Gerenciamento da Tela:

- **Ctrl + L**: Limpa a tela do terminal, mantendo a linha de comando atual no topo.

3. Atalhos Específicos do PowerShell

O PowerShell é um shell de linha de comando e linguagem de script desenvolvido pela Microsoft, focado em automação e gerenciamento de configurações. Muitos de seus atalhos são influenciados pelo ambiente Windows e pelo módulo PSReadLine.

3.1. Edição de Texto e Comando

- **Ctrl + Home**: Move o cursor para o início do buffer de entrada (início da linha).
- **Ctrl + End**: Move o cursor para o final do buffer de entrada (final da linha).
- **Ctrl + Left Arrow**: Move o cursor uma palavra para trás.
- **Ctrl + Right Arrow**: Move o cursor uma palavra para frente.
- **Ctrl + Backspace**: Apaga a palavra anterior ao cursor.
- **Ctrl + Delete**: Apaga a palavra seguinte ao cursor.
- **Ctrl + Space (No ISE/VS Code com extensão)**: Exibe a ajuda do IntelliSense (preenchimento automático de comandos e parâmetros).
- **Ctrl + Y**: Em PowerShell ISE, refaz a última ação desfeita (“Redo”). No console, com o módulo PSReadLine, pode funcionar como “Yank” (colar texto previamente “morto” ou “recortado” para o kill ring).
- **Ctrl + Z**: Em PowerShell ISE, desfaz a última ação. No console, pode suspender um processo, similar ao Bash.

3.2. Controle de Execução

- **Enter**: Executa o comando atual.
- **Shift + Enter**: Adiciona uma nova linha sem executar o comando (útil para comandos de várias linhas).

3.3. Histórico de Comandos

- **Seta para Cima (↑)**: Exibe o comando anterior no histórico.
- **Seta para Baixo (↓)**: Exibe o próximo comando no histórico.

- **F7**: Exibe uma janela com o histórico de comandos, permitindo selecionar com as setas.
- **Alt + F7**: Limpa o histórico de comandos da sessão atual.
- **F8**: Pesquisa o histórico de comandos a partir da posição atual do cursor.
- **F9**: Seleciona um comando do histórico pelo seu número.
- **Alt + .**: Insere o último argumento do comando anterior.
- **Ctrl + R**: Pesquisa incremental inversa no histórico de comandos (com PSReadLine).

3.4. Interface do PowerShell ISE (Integrated Scripting Environment)

O ISE é um ambiente de desenvolvimento integrado para PowerShell, com recursos gráficos e de edição avançados.

- **F1**: Abre a ajuda para o cmdlet ou tópico atual.
- **F5**: Executa o script completo no painel de script.
- **F8**: Executa a seleção atual no painel de script.
- **Ctrl + N**: Abre um novo painel de script.
- **Ctrl + O**: Abre um arquivo existente.
- **Ctrl + S**: Salva o arquivo atual.
- **Ctrl + M**: Expande ou recolhe regiões de código (code folding).
- **Ctrl + F**: Abre a caixa de diálogo “Localizar” no painel de script.
- **Ctrl + H**: Abre a caixa de diálogo “Substituir” no painel de script.
- **Ctrl + J**: Mostra snippets de código (modelos de código pré-definidos).
- **Ctrl + Shift + A**: Alterna o comentário de bloco (<# ... #>).
- **Ctrl + K**: Alterna o comentário de linha (#).

4. Comandos Comuns e Específicos para Bash, CMD e PowerShell

Esta seção oferece uma comparação de comandos essenciais em três dos ambientes de linha de comando mais utilizados. Embora a funcionalidade seja similar, a sintaxe e a filosofia de cada shell podem variar.

5. Comandos Comuns (funcionalidade similar ou equivalentes)

Aqui estão os comandos que executam funções semelhantes ou idênticas em todos os três ambientes, embora a sintaxe exata possa variar ligeiramente.

5.1. Navegação de Diretórios:

- **Bash/CMD/PowerShell**: `cd <diretório>`
– Muda o diretório de trabalho atual.

- Ex: `cd Documentos` (Windows) ou `cd ~/Documentos` (Linux/macOS).
- `cd ..`: Volta um nível no diretório.
- `cd ~` (Bash/PowerShell): Vai para o diretório home do usuário.
- `cd -` (Bash): Volta para o diretório anterior.

5.2. Listar Conteúdo do Diretório

- **Bash/PowerShell:** `ls`
 - Lista o conteúdo do diretório atual.
 - `ls -l`: Lista em formato longo (detalhes).
 - `ls -a`: Lista incluindo arquivos ocultos.
- **CMD/PowerShell:** `dir`
 - Lista o conteúdo do diretório atual.
 - `dir /w`: Lista em formato largo.
 - `dir /a`: Lista incluindo arquivos ocultos.

5.3. Criar Diretório

- **Bash/CMD/PowerShell:** `mkdir <nome_do_diretório>`
 - Cria um novo diretório.
 - Ex: `mkdir MeusProjetos`
 - `mkdir -p <caminho/para/diretório>` (Bash): Cria diretórios pais se não existirem.

5.4. Remover Arquivo

- **Bash:** `rm <nome_do_arquivo>`
- **CMD:** `del <nome_do_arquivo>`
- **PowerShell:** `Remove-Item <nome_do_arquivo>` (ou `rm <nome_do_arquivo>` como alias)

5.5. Remover Diretório (Vazio)

- **Bash/CMD:** `rmdir <nome_do_diretório>`
- **PowerShell:** `Remove-Item <nome_do_diretório>` (ou `rmdir <nome_do_diretório>` como alias)

5.6. Remover Diretório (Não Vazio/Recursivo)

- **Bash:** `rm -r <nome_do_diretório>`
 - O `-r` significa recursivo. Use com cautela, pois não pede confirmação por padrão.
- **CMD:** `rmdir /s <nome_do_diretório>`
 - O `/s` significa recursivo. Geralmente pede confirmação. Para forçar sem confirmação, adicione `/q` (`rmdir /s /q`).
- **PowerShell:** `Remove-Item -Recurse <nome_do_diretório>`
 - O `-Recurse` remove o diretório e seu conteúdo. Para forçar sem confirmação, adicione `-Force`.

5.7. Copiar Arquivos

- **Bash:** `cp <origem> <destino>`
 - Ex: `cp arquivo.txt /tmp/`
- **CMD:** `copy <origem> <destino>`
 - Ex: `copy arquivo.txt C:\Temp\`
- **PowerShell:** `Copy-Item <origem> <destino>` (ou `cp <origem> <destino>` como alias)
 - Ex: `Copy-Item arquivo.txt C:\Temp\`

5.8. Mover/Renomear Arquivos

- **Bash:** `mv <origem> <destino>`
 - Move um arquivo ou diretório. Se o destino for um novo nome no mesmo diretório, renomeia.
- **CMD:**
 - **Mover:** `move <origem> <destino>`
 - **Renomear:** `ren <origem> <novo_nome>`
- **PowerShell:**
 - **Mover:** `Move-Item <origem> <destino>` (ou `mv <origem> <destino>` como alias)
 - **Renomear:** `Rename-Item <origem> <novo_nome>`

5.9. Exibir Conteúdo de Arquivo

- **Bash:** `cat <nome_do_arquivo>`
 - Exibe o conteúdo completo do arquivo. Para arquivos grandes, use `less <nome_do_arquivo>` para visualização paginada.
- **CMD:** `type <nome_do_arquivo>`
- **PowerShell:** `Get-Content <nome_do_arquivo>` (ou `cat <nome_do_arquivo>` como alias)

5.10. Limpar Tela

- **Bash/PowerShell:** `clear`
- **CMD/PowerShell:** `cls`

5.11. Testar Conectividade de Rede

- **Bash/CMD/PowerShell:** `ping <endereço_ip_ou_dominio>`
 - Envia pacotes ICMP para testar a conectividade com um host.

5.12. Configuração de IP (Exibir)

- **Bash:** `ip a` (moderno) ou `ifconfig` (em sistemas mais antigos)
 - Exibe informações sobre interfaces de rede e endereços IP.
- **CMD/PowerShell:** `ipconfig`
 - Exibe informações de configuração de IP para adaptadores de rede no Windows.

5.13. Sair do Terminal

- **Bash/CMD/PowerShell:** `exit`
 - Encerra a sessão atual do shell.

6. Comandos Específicos do Bash

O Bash é conhecido por sua versatilidade e integração com ferramentas poderosas do ecossistema Unix/Linux.

- **grep <padrão> <arquivo>:** Filtra linhas em um arquivo que correspondem a um padrão.
Ex: `grep "erro" log.txt`

- **awk '{print \$1}' <arquivo>**: Ferramenta de processamento de texto para extrair e manipular dados em colunas. Ex: `cat dados.txt | awk '{print $1, $3}'`
- **sed 's/antigo/novo/g' <arquivo>**: Editor de fluxo para transformar texto. Ex: `sed 's/foo/bar/g' arquivo.txt`
- **ssh <usuário>@<host>**: Conecta-se a um servidor remoto via Secure Shell.
- **sudo <comando>**: Executa um comando com privilégios de superusuário.
- **chmod <permissões> <arquivo>**: Altera as permissões de acesso de um arquivo. Ex: `chmod +x script.sh` (torna executável).
- **chown <usuário>:<grupo> <arquivo>**: Altera o proprietário e/ou grupo de um arquivo.
- **tar -czvf <arquivo.tar.gz> <diretório>**: Cria ou extrai arquivos compactados (tarballs).
 - -c: criar, -z: gzip, -v: verbose, -f: arquivo.
 - Para extrair: `tar -xzvf <arquivo.tar.gz>`.
- **ps aux**: Exibe informações sobre processos em execução.
- **top**: Exibe um monitor de processos dinâmico em tempo real.
- **history**: Exibe o histórico de comandos digitados.
- **man <comando>**: Exibe o manual (página de ajuda) de um comando.
- **df -h**: Exibe o uso do disco em formato legível (human-readable).
- **du -sh <diretório>**: Exibe o tamanho total de um diretório em formato legível.
- **find <caminho> -name <nome>**: Procura arquivos e diretórios. Ex: `find . -name "*.log"`
- **wget <URL> / curl <URL>**: Ferramentas para baixar arquivos ou fazer requisições HTTP.

7. Comandos Específicos do CMD (Prompt de Comando do Windows)

O CMD é o shell tradicional do Windows, focado em operações de sistema de arquivos e comandos básicos do Windows.

- **tasklist**: Exibe uma lista de processos em execução.
- **taskkill /IM <nome_do_processo.exe> /F**: Finaliza um processo em execução. O /F força o encerramento.
- **netstat -ano**: Exibe conexões de rede ativas, portas de escuta e IDs de processo.
- **reg query <chave>**: Consulta chaves e valores do Registro do Windows.
- **schtasks /create /tn <nome_da_tarefa> /tr <comando_a_executar> /sc daily**: Cria tarefas agendadas.
- **systeminfo**: Exibe informações detalhadas sobre o sistema operacional e hardware.
- **diskpart**: Utilitário de linha de comando para gerenciamento de partições de disco (requer privilégios de administrador).

- **tree <caminho>**: Exibe a estrutura de diretórios de forma gráfica.
- **gpupdate /force**: Atualiza as políticas de grupo.
- **sfc /scannow**: Verifica e repara arquivos de sistema protegidos.
- **chkdsk**: Verifica e repara erros no sistema de arquivos de um volume.
- **format**: Formata um disco.

8. Comandos Específicos do PowerShell

O PowerShell é um shell orientado a objetos com uma estrutura de “cmdlets” (command-lets) que oferecem funcionalidades avançadas de gerenciamento e automação.

- **Get-Process**: Lista os processos em execução (equivalente a ps no Bash, mas retorna objetos que podem ser manipulados).
- **Stop-Process -Name "<nome_do_processo>"**: Interrompe um ou mais processos.
- **Get-Service**: Lista os serviços do Windows e seu status.
- **Set-Service -Name "<nome_do_serviço>" -Status Running**: Inicia, para ou reinicia um serviço.
- **Invoke-WebRequest -Uri <URL>**: Faz requisições HTTP para buscar conteúdo web.
- **New-Item -Path <caminho> -ItemType File -Name <nome_do_arquivo>**: Cria novos arquivos ou diretórios.
 – Ex: `New-Item -Path "C:\Temp\novo.txt" -ItemType File`
- **Remove-Item -Path <caminho>**: Remove um arquivo ou diretório.
- **Copy-Item -Path <origem> -Destination <destino>**: Copia itens (arquivos ou diretórios).
- **Move-Item -Path <origem> -Destination <destino>**: Move itens.
- **Select-Object -Property <propriedade1>, <propriedade2>**: Seleciona propriedades de objetos.
 – Ex: `Get-Process | Select-Object Name, Id, CPU`
- **Where-Object -Property <propriedade> -EQ <valor>**: Filtra objetos com base em propriedades.
 – Ex: `Get-Service | Where-Object Status -EQ "Running"`
- **ForEach-Object { <script_block> }**: Processa cada objeto em uma coleção.
 – Ex: `Get-Service | ForEach-Object { Write-Host $_.Name }`
- **Get-Help <cmdlet>**: Obtém ajuda detalhada sobre um cmdlet.
- **Get-Command**: Lista todos os comandos (cmdlets, funções, aliases, etc.) disponíveis.

- **Set-ExecutionPolicy RemoteSigned:** Define a política de execução de scripts. É crucial para permitir a execução de scripts baixados da internet (mas exige que sejam assinados digitalmente).
- **Get-ChildItem:** Equivalente a `ls` ou `dir`, lista o conteúdo de diretórios.

9. Gerenciamento de Variáveis de Ambiente

Variáveis de ambiente são pares nome-valor que armazenam informações sobre o ambiente de execução do sistema. Elas são usadas por programas e scripts para configurar seu comportamento.

- **Bash:**
 - **Listar todas:** `printenv` ou `env`
 - **Exibir uma específica:** `echo $NOME_DA_VARIAVEL`
 - **Definir (temporário para a sessão):** `export NOME_DA_VARIAVEL="valor"`
 - **Definir (permanente):** Adicione a linha `export NOME_DA_VARIAVEL="valor"` ao arquivo `~/.bashrc` ou `~/.profile` e recarregue com `source ~/.bashrc`.
 - **Remover:** `unset NOME_DA_VARIAVEL`
- **CMD:**
 - **Listar todas:** `set`
 - **Exibir uma específica:** `echo %NOME_DA_VARIAVEL%`
 - **Definir (temporário para a sessão):** `set NOME_DA_VARIAVEL=valor`
 - **Definir (permanente):** `setx NOME_DA_VARIAVEL "valor"` (cria uma variável de ambiente de usuário permanente, disponível em novas sessões do Prompt de Comando ou PowerShell, e após reiniciar o sistema para algumas aplicações).