

# Linux: Comandos úteis (Guia de bolso)



## Arquivos e diretórios

### LISTAR ARQUIVOS CLASSIFICADOS POR EXTENSÃO

```
ls -l -X  
# Or  
ls -l --sort=extension
```

### CONTAR O NÚMERO DE ARQUIVOS EM UM DIRETÓRIO

```
ls -l ./my-directory | wc -l
```

*Note: use `-a` to include also dot files.*

### CRIE UM LINK SIMBÓLICO PARA UM DIRETÓRIO OU ARQUIVO

```
ln -s /var/www/html ~/www  
ln -s ~/my/large/path/file.txt ~/myfile.txt
```

### ALTERAR PERMISSÕES APENAS PARA UM LINK SIMBÓLICO

```
chmod -h 770 ~/www  
chown -h www-data:www-data ~/www
```

## ENCONTRE AS PERMISSÕES OCTAIS DE UM ARQUIVO OU DIRETÓRIO

```
stat -c "%a %n" /var/www/html  
# 770 /var/www/html
```

## ADICIONAR PERMISSÕES FIXAS A UM DIRETÓRIO E SUBDIRETÓRIOS

```
find /var/www/html -type d -exec chmod g+s {} \
```

*Nota: Qualquer novo arquivo criado terá as mesmas permissões que a pasta raiz*

## CRIE UM ARQUIVO VAZIO NO DIRETÓRIO E SUBDIRETÓRIOS ATUAIS

```
find ./my/current/directory -type d -exec touch {}/.gitignore \;
```

## EXCLUA UM ARQUIVO ESPECÍFICO DO DIRETÓRIO ATUAL E SUBDIRETÓRIOS

```
find ./my/current/directory -name ".gitignore" -type f -delete
```

## EXCLUIR TODO O CONTEÚDO DE UM DIRETÓRIO, EXCETO UM EM ESPECÍFICO

```
find ./my-directory -mindepth 1 ! -regex '^./my-directory/data\(\/.*\)?' -  
delete
```

## COPIAR O CONTEÚDO DO ARQUIVO PARA A ÁREA DE TRANSFERÊNCIA

Copiar a saída do comando shell para a área de transferência

```
cat myfile.txt | xclip -selection c
```

## COPIAR DIRETÓRIO INTEIRO PARA O DESTINO

```
cp -avr /my/current/directory /destination/directory  
# Or  
rsync -av /my/current/directory /destination/directory
```

## COPIE TODOS OS ARQUIVOS DO DIRETÓRIO APENAS PARA O DESTINO COM A OPÇÃO DE EXCLUSÃO

```
rsync -har --progress --exclude .git /current/directory/.  
/destination/directory
```

## TRANSFERIR TODO O CONTEÚDO DE UM DIRETÓRIO PARA UM DESTINO REMOTO

\_\_Nota: \_\_ É necessário configurar uma conexão SSH. Você pode usar um [arquivo de configuração SSH](#).

```
rsync -hrtpluz --progress --stats my-current-directory/. my-remote-  
server:/var/data/my-remote-dir
```

Opções usadas acima:

-h,	números de saída em um formato legível
-r	percorrer diretórios
-t, --times	preservar os tempos de modificação
-p, --perms	preservar permissões
-l, --links	copiar links simbólicos como links simbólicos
-u, --update	pular arquivos que são mais recentes no receptor
-z, --compress	compactar os dados do arquivo durante a transferência

--progress	mostrar o progresso durante a transferência
--stats	dar algumas estatísticas de transferência de arquivos

## MOSTRA O USO DE ESPAÇO DO ARQUIVO OU DIRETÓRIO

Mostra o uso de espaço do arquivo ou diretório (recursivo) em formato legível por humanos.

```
du -sh /var/log/dnf.librepo.log
# 4,1M /var/log/dnf.librepo.log
```

```
du -sh /var/log
# 2,2G /var/log
```

## MOSTRA O TAMANHO DE UM LINK SIMBÓLICO (ARQUIVO OU DIRETÓRIO)

```
du -Hsh $(which dart)
# 21M /usr/local/bin/dart
```

## MOSTRA O USO DE ESPAÇO DE DIRETÓRIOS E ARQUIVOS ORDENADOS POR TAMANHO

```
du -sh /var/data/software/app/* | sort -rh
# 1.1G /var/data/software/app/backups
# 223M /var/data/software/app/logs
# 125M /var/data/software/app/attachments
# 70M /var/data/software/app/recovery
# 1.2M /var/data/software/app/shared-home
# 592K /var/data/software/app/viewfile
# 12K /var/data/software/app/journal
# 4.0K /var/data/software/app/bundled-plugins
# 0 /var/data/software/app/lock
```

## MOSTRA OS DEZ PRINCIPAIS DOS MAIORES DIRETÓRIOS

```
du -hS /home/user/some/directory | sort -rh | head -10
# 90G    /home/user/some/directory/big-dir
# 10G    /home/user/some/directory/subdir/another-big-dir
# ...
```

## EXCLUA TODOS OS ARQUIVOS NO DIRETÓRIO POR PADRÃO

```
find /usr/local/apache/logs/archive/ -name '*2017.gz' -delete
```

## MOVER ARQUIVOS POR PADRÃO

*This command move all \*.js files into \*.ts files (move equivalent)*

```
find src/ -type f -name "*.js" -exec bash -c 'mv {} `echo {} | sed -e "s/.js/.ts/g"`' \;
```

## COMPACTAR ARQUIVOS POR PADRÃO USANDO TAR E GZIP

O comando abaixo comprime um grupo de arquivos por padrão usando compactação tar e gzip em um novo arquivo como `FILE_NAME.tar.gz`. Nenhuma classificação garantida.

```
# Input files:
# src/file1.log.01
# src/file1.log.02
# src/file1.log.03
```

```
find src/ -type f -name "file1.log.*" -exec bash -c "tar cf - {} | gzip -9 > {}.tar.gz" \;
```

```
# Or using find -exec
```

```
find src/ -maxdepth 1 -type f -name "file1.log.*" -exec sh -c "tar cf - {} | gzip -9 > tarballs/{}.tar.gz && echo '{} (compressed)'" \;
```

```
# Output files:
# src/file1.log.01.tar.gz
```

```
# src/file1.log.02.tar.gz  
# src/file1.log.03.tar.gz
```

## OTIMIZANDO O TAMANHO DO ARQUIVO TAR / GZIP ALTERANDO A ORDEM DOS ARQUIVOS DE CONTEÚDO

```
find directory/* -print | rev | sort | rev | \  
    tar --create --no-recursion --files-from - | \  
    gzip -c > directory.tar.gz
```

Motivado por: <https://news.ycombinator.com/item?id=24221498>

### Classificação alternativa de forma crescente:

```
find . -type f -name "file1.log.*" | sort -n | xargs -I{} sh -c "tar cf  
- {} | gzip -9 > {}.tar.gz && echo '{} (compressed)'" \;
```

**Dica:** Use o parâmetro `-maxdepth` para limitar a localização dos níveis de diretório.

## COMPACTAR UM DIRETÓRIO USANDO TAR E GZIP

```
tar -zcvf my_dir.tar.gz my-dir/
```

## LISTA O CONTEÚDO DE UM ARQUIVO TAR E GZIP SEM EXTRAÍ-LO

```
tar -tvf my_dir.tar.gz | less
```

## CRIE UM DIRETÓRIO TEMPORÁRIO COM PREFIXO

```
mktemp -dt "my-prefix"
```

## LIMPAR DIRETÓRIO TEMPORÁRIO

```
rm -rf /tmp/* /tmp/.*
```

## CALCULAR O TAMANHO GZIP DE UM ARQUIVO NÃO COMPACTADO

```
gzip -c FILENAME.txt | wc -c | awk '{
  if ($1 > 1000 ^ 3) {
    print($1 / (1000 ^ 3)"G")
  } else if ($1 > 1000 ^ 2) {
    print($1 / (1000 ^ 2)"M")
  } else if ($1 > 1000) {
    print($1 / 1000"K")
  } else {
    print($1)"b"
  }
}'
```

```
# 560K
```

## DIVIDA UM ARQUIVO ESPECÍFICO EM MAIS ARQUIVOS PEQUENOS POR NÚMERO DE LINHAS

```
split -d -l 200000 ./big_file.log small_file.log.
```

```
# Output files in current directory:
# small_file.log.01
# small_file.log.02
# small_file.log.03
# ....
```

*Notas:*

- **smallfile.log** é um sufixo personalizado para nomes de arquivo.
- **200000** é um número personalizado de linhas por arquivo.

## OBTENHA APENAS O NÚMERO DE LINHAS DE UM ARQUIVO ESPECÍFICO

```
wc -l big_file.log | sed -E "s/([a-z\-\_\.]|[:space:])*//g"
```

```
# 9249133
```

## Caminhos

### MOSTRA O CAMINHO COMPLETO DE UM COMANDO

```
which bash  
# /usr/bin/bash
```

```
which git node  
# /usr/bin/git  
# /usr/bin/node
```

### MOSTRA O CAMINHO RESOLVIDO DE UM LINK SIMBÓLICO

```
realpath ~/www  
# /usr/share/nginx/html
```

### DETERMINE O DIRETÓRIO ATUAL

```
pwd  
# /home/my/current/directory
```

## Dispositivos

### EXIBIR O USO DE ESPAÇO EM DISCO DO SISTEMA DE ARQUIVOS COM TOTAL

Mostra o uso de espaço em disco do sistema de arquivos em formato legível por humanos.



```
df -h --total
```

# Filesystem	Size	Used	Avail	Use%	Mounted on
# devtmpfs	487M	0	487M	0%	/dev
# tmpfs	497M	0	497M	0%	/dev/shm
# tmpfs	497M	508K	496M	1%	/run
# tmpfs	497M	0	497M	0%	/sys/fs/cgroup
# /dev/vda1	30G	2.7G	26G	10%	/
# tmpfs	100M	0	100M	0%	/run/user/0
# total	2.2T	600G	100G	20%	-

## EXIBIR INFORMAÇÕES DE MEMÓRIA DO SISTEMA COM TOTAL

```
free -h --total
```

#	total	used	free	shared	buff/cache
available					
# Mem:	200G	60G	100G	262M	30G
180G					
# Swap:	0B	0B	0B		
# Total:	200G	60G	100G		

or

```
cat /proc/meminfo
```

```
# MemTotal:      183815530 kB
# MemFree:       101918660 kB
# MemAvailable:  123712410 kB
# ....
```

Dica: Pipe **grep** para filtrar seus resultados. Por exemplo, `cat/proc/meminfo | grep MemTotal`

## MONTE UM DISPOSITIVO USB FAT32

```
mount -t vfat /dev/sdb1 /media/usb
```

## AUMENTAR O TAMANHO DO DIRETÓRIO TEMPORÁRIO

```
mount -o remount,size=5G /tmp/
```

## DESCUBRA UM DISPOSITIVO DE BLOCO POR RÓTULO

```
blkid --label MY_MICROSD  
# /dev/sdc1
```

```
lsblk --output name,serial,uuid,partuuid,label,partlabel $(blkid --label  
MY_MICROSD)
```

#	NAME	SERIAL	UUID	PARTUUID	LABEL	PARTLABEL
#	sdc1		4BGE-7510	ec123nba-02	MY_MICROSD	

## Usuários e grupos

### ALTERNE O USUÁRIO E EXECUTE O COMANDO IMEDIATAMENTE

```
sudo -Hu root fish
```

### ADICIONAR UM USUÁRIO EXISTENTE A UM GRUPO EXISTENTE

```
usermod -a -G ftp john
```

## Data hora

### MOSTRAR DATA NO FORMATO ISO ESTENDIDO ([ISO 8601](#))

```
date "+%Y-%m-%dT%H:%m:%S"  
# 2018-09-13T10:09:26
```

## Rede

## **MOSTRAR ENDEREÇO IP ATUAL**

```
ifconfig | awk '/<UP,BROADCAST,RUNNING,MULTICAST>/ { getline; print $2 }'  
# or  
ifconfig | grep -E "([0-9]{1,3}\.){3}[0-9]{1,3}" | grep -v 127.0.0.1 | awk  
'{ print $2 }' | cut -f2 -d: | head -n1
```

## **Diversos**

### **MOSTRAR DISTRIBUIÇÃO GNU / LINUX E SISTEMA OPERACIONAL**

a) Using `hostnamectl`:

```
hostnamectl | grep -E "(.+) (System|Kernel|Arch) (.+)"  
# Operating System: Arch Linux  
# Kernel: Linux 5.1.7-arch1-1-ARCH  
# Architecture: x86-64
```

b) Using `cat /etc/os-release`:

```
cat /etc/os-release  
# NAME="Arch Linux"  
# PRETTY_NAME="Arch Linux"  
# ID=arch  
# BUILD_ID=rolling  
# ANSI_COLOR="0;36"  
# HOME_URL="https://www.archlinux.org/"  
# DOCUMENTATION_URL="https://wiki.archlinux.org/"  
# SUPPORT_URL="https://bbs.archlinux.org/"  
# BUG_REPORT_URL="https://bugs.archlinux.org/"  
# LOGO=archlinux
```

### **GERAR NÚMEROS ALEATÓRIOS**

a)

```
od -vAn -N64 < /dev/urandom | tr '\n' ' ' | sed "s/ //g" | head -c 32  
# 03121617301002504516642404031105
```

b)

```
env LC_CTYPE=C tr -dc "0-9" < /dev/urandom | head -c 32 | xargs  
# 50569696992247151969921987764342
```

Change *head* value to truncate the result's length.

## GERAR ALFANUMÉRICOS ALEATÓRIOS

a) Apenas alfanumérico

```
base64 /dev/urandom | tr -d '/+' | head -c 32 | xargs  
# 3udiq6F74a1wcPwXzIDWSnjRYQXcxiyl
```

b) Alfanumérico com um conjunto de caracteres personalizados

```
env LC_CTYPE=C tr -dc "A-Za-z0-9_!@#\$%^&*()-+=" < /dev/urandom | head -c  
32 | xargs  
# yiMg^Cha=Zh$6Xh%zDQAYBH1SI6Po(&P
```

Change *tr -dc* char set to get a custom result.

## GERAR UM HASH ALEATÓRIO

```
od -vAn -N64 < /dev/urandom | tr '\n' ' ' | sed "s/ //g" | openssl dgst -  
sha256 | sed "s/-//g"  
# 7adf57e0a90b32ce0e1f446268dbd62b583c649a2e71a426519c6e9c0006b143
```

Openssl digest algorithms supported: *md5*, *md4*, *md2*, *sha1*, *sha*, *sha224*, *sha256*, *sha384*,  
*sha512*, *mdc2* and *ripemd160*

## GERAR UM UUID ALEATÓRIO

```
uuidgen | tr "[:lower:]" "[:upper:]"  
# D2DA7D0C-ABAA-4866-9C97-61791C9FEC89
```

## GERE 1 MILHÃO DE NÚMEROS DE TELEFONE ALEATÓRIOS EXCLUSIVOS

Este comando gera um milhão de números de telefone aleatórios únicos (permutações aleatórias) rapidamente usando GNU/Linux com o comando [shuf](#).

Use o comando **sed** para personalizar cada formato de número. Por exemplo, para adicionar algum prefixo ou sufixo. Lembre-se de que **shuf** não se limita apenas a números.

```
shuf -i 100000000-999999999 -n 1000000 | sed -e 's/^/51/' > gsm.txt
```

## BAIXE UM ARQUIVO USANDO CURL E OBTENHA SUAS ESTATÍSTICAS

Os comandos abaixo baixam um arquivo usando Curl e obtêm suas estatísticas de download em um arquivo `curl-download-stats.log`. Coloque o conteúdo do script em um arquivo personalizado `.sh` e execute apenas ele.

```
#!/bin/bash
```

```
curl -Skw "\n\
CURL download testing\n\
=====\n\
URL:                %{url_effective}\n\
Response code:      %{response_code}\n\
Download size:       %{size_download}B\n\
Download speed:      %{speed_download}B/s\n\
Time connect:        %{time_connect}s\n\
Time name lookup:    %{time_namelookup}s\n\
Time pretransfer:    %{time_pretransfer}\n\
Time start transfer: %{time_starttransfer}s\n\
Time redirect:       %{time_redirect}s\n\
Time total:          %{time_total}s\n" \
-Lo /dev/null --url \
https://ger.mirror.pkgbuild.com/iso/2020.02.01/archlinux-2020.02.01-
```

```
x86_64.iso \  
> curl-download-stats.log
```

### Notas:

- Para colocar o arquivo em um local, basta substituir `/dev/null` pelo caminho do arquivo correspondente.
- Para personalizar o formato de gravação, dê uma olhada em <https://ec.haxx.se/usingcurl/usingcurl-verbose/usingcurl-writeout>



2021 [Reativa Tecnologia](#)