

# REFERÊNCIA RÁPIDA FIND

Busca de arquivos por nome, tipo, tamanho, tempo, permissões e ações

## Busca Básica

### Executando find

```
find . # list all files recursively
find /var/log # search from specific path
find . -name "*.txt" # find by name
find / -name "config" 2>/dev/null # suppress permission errors
find dir1 dir2 -name "*.py" # search multiple directories
```

### Visão Geral da Sintaxe

**find [path...] [tests] [actions]** Forma geral — caminhos, testes, depois ações

**path** Diretório inicial (padrão: diretório atual)

**test** Condição para filtrar arquivos (-name, -type, etc.)

**action** O que fazer com as correspondências (-print, -exec, etc.) -print se nenhuma ação for especificada

**Default action**

**Padrões de Nome**

### Correspondência de Nome

```
find . -name "*.log" # case-sensitive glob
find . -iname "readme*" # case-insensitive glob
find . -name "*.py" -o -name "*.js" # OR: Python or JS files
find . -path "*/src/*.ts" # match against full path
find . -regex '.*\.(py|js|)' # POSIX regex on full path
```

### Testes de Nome

**-name pattern** Basename corresponde a glob shell (case-sensitive)

**-iname pattern** Basename corresponde a glob (case-insensitive)

**-path pattern** Caminho completo corresponde a glob shell

**-ipath pattern** Caminho completo corresponde a glob (case-insensitive)

**-regex pattern** Caminho completo corresponde a expressão regular

**-iregex pattern** Caminho completo corresponde a regex (case-insensitive)

### Filtros de Tipo

#### Filtrar por Tipo

```
find . -type f # regular files only
find . -type d # directories only
find . -type l # symbolic links
find . -type f -name "*.sh" # combine type + name
```

### Tipos de Arquivo

**-type f** Arquivo regular

**-type d** Diretório

**-type l** Link simbólico

**-type b** Dispositivo de bloco

**-type c** Dispositivo de caractere

**-type p** Pipe nomeado (FIFO)

**-type s** Socket

**-empty** Arquivo ou diretório vazio

### Tamanho e Tempo

#### Exemplos de Tamanho e Tempo

```
find . -size +100M # larger than 100 MB
find . -size -1k # smaller than 1 KB
find . -mtime 7 # modified in last 7 days
find . -mmin 30 # modified in last 30 minutes
find . -newer reference.txt # newer than reference file
```

### Testes de Tamanho e Tempo

**-size +/-NC** Tamanho em bytes (c), kilobytes (k), megabytes (M), gigabytes (G)

**-mtime +/-N** Modificado há N\*24 horas (+mais antigo, -mais recente)

**-atime +/-N** Acessado há N\*24 horas

**-ctime +/-N** Status alterado há N\*24 horas

**-mmin +/-N** Modificado há N minutos

**-newer file** Modificado mais recentemente que o arquivo

**-newermt date** Modificado após string de data (GNU)

### Permissões

#### Exemplos de Permissão

```
find . -perm 644 # exact permissions: rw-r--r--
find . -perm -u+x # user has execute bit set
find . -perm /o+w # others have write (any match)
find . -user root # owned by root
find . -group www-data -type f # owned by group
```

### Testes de Permissão

**-perm mode** Correspondência exata de permissão

**-perm -mode** Todos os bits especificados estão definidos

**-perm /mode** Algum bit especificado está definido

**-user name** Pertence ao usuário (nome ou UID)

**-group name** Pertence ao grupo (nome ou GID)

**-nouser** Nenhum usuário correspondente em /etc/passwd

**-nogroup** Nenhum grupo correspondente em /etc/group

### Ações

#### Exemplos de Ação

```
find . -name "*.log" -print # print paths (default)
find . -name "*.tmp" -delete # delete matching files
find . -type f -ls # detailed listing
find . -name "*.txt" -print0 # null-delimited output
find . -type f -printf "%p %s\n" # custom format (GNU)
```

### Referência de Ações

**-print** Imprimir caminho (delimitado por nova linha)

**-print0** Imprimir caminho (delimitado por null, seguro para xargs)

**-ls** Imprimir detalhes do arquivo (como `ls -dils`)

**-delete** Excluir arquivos correspondentes (implica -depth)

**-printf format** Formato de saída personalizado (GNU): `%p` caminho, `%s` tamanho, `%t` hora

**-fprint file** Escrever caminhos em arquivo

**-quit** Sair após primeira correspondência

### Combinando Testes

#### Operadores Lógicos

```
find . -name "*.py" -type f # implicit AND
find . -name "*.py" -a -size +10k # explicit AND
find . -name "*.py" -o -name "*.js" # OR
find . ! -name "*.log" # NOT
find . \( -name "*.py" -o -name "*.js" \) -type f
```

### Referência de Operadores

**expr1 expr2 / expr1 -a expr2** AND — ambos devem ser verdadeiros (padrão)

**expr1 -o expr2** OR — um deve ser verdadeiro

**! expr / -not expr** NOT — negar a expressão

**\( expr \)** Agrupar expressões (escapar parênteses no shell)

**Evaluation order** Da esquerda para direita; -a tem precedência sobre -o

### Exec e Delete

#### Exemplos de Exec

```
find . -name "*.sh" -exec chmod +x {} \;
find . -name "*.log" -exec rm {} +
find . -type f -exec grep -l "TODO" {} +
find . -name "*.bak" -ok rm {} \; # prompt before each
find . -name "*.tmp" -print0 | xargs -0 rm
```

### Referência de Exec

**-exec cmd {} \;** Executar cmd uma vez por arquivo ({} = caminho)

**-exec cmd {} +** Executar cmd com múltiplos arquivos de uma vez (mais rápido)

**-ok cmd {} \;** Como -exec mas solicita confirmação

**-execdir cmd {} \;** Executar cmd a partir do diretório do arquivo

**xargs -0** Combinar com -print0 para processamento seguro em lote

**-delete** Excluir arquivos; processar os mais profundos primeiro

### Profundidade e Poda

#### Exemplos de Profundidade e Poda

```
find . -maxdepth 1 -type f # current dir only
find . -mindepth 2 -name "*.py" # skip top-level
find . -name ".git" -prune -o -print # skip .git dirs
find . -depth -name "*.tmp" -delete # process children first
```

### Opções de Profundidade

**-maxdepth N** Descer no máximo N níveis (0 = apenas caminho inicial)

**-mindepth N** Não aplicar testes em níveis menores que N

**-depth** Processar conteúdo do diretório antes do próprio diretório

**-prune** Não descer no diretório correspondente

**-mount / -xdev** Não cruzar fronteiras de sistema de arquivos

**-follow / -L** Seguir links simbólicos

### Padrões Comuns

#### One-Liners

```
find . -name "*.pyc" -delete # clean Python bytecode
find . -type f -size 0 -delete # remove empty files
find . -mtime +30 -name "*.log" -delete # purge old logs
find . -type f -name "*.md" | wc -l # count Markdown files
find . -type d -empty -delete # remove empty dirs
```

### Receitas

**Find largest files** `find . -type f -printf "%s %p\n" | sort -rn | head`

**Find duplicates by name** `find . -type f | awk -F/ '{print \$NF}' | sort | uniq -d`

**Rename extension** `find . -name "\*.txt" -exec rename 's/.txt/.md/' {} +`

**Find broken symlinks** `find . -xtype l`

**Archive recent files** `find . -mtime -7 -print0 | tar czf recent.tar.gz --null -T`

**Search code files** `find . -name "\*.py" -exec grep -l 'pattern' {} +`