

# Linux OS

## Exemplos do comando "find" no Linux

No Linux, o comando **find** permite fazer pesquisas de determinados ficheiros ou pastas dentro do sistema de ficheiros.

Permite também conjugar vários comandos para executar determinadas tarefas.

Vamos mostrar alguns exemplos concretos da utilização do comando **find**.

### Usar o comando find no Linux com base no nome do ficheiro

Imagine que você precisa de procurar um ficheiro de imagem com o nome `articles.jpg`, dentro duma instalação Joomla, alojada num servidor Linux. Na document root, execute o comando:

```
find . -name "articles.jpg"
```

#### OUTPUT

```
./images/stories/articles.jpg
```

se quisermos também executar algum comando no ficheiro, como, por exemplo, alterar as permissões do ficheiro:

```
find . -name "articles.jpg" -exec chmod 644 {} \;
```

Se quiser por exemplo pesquisar o mesmo nome de ficheiro dentro da pasta `images`:

```
find images -name "articles.jpg"
```

# Linux OS

Se preferir usar um caminho físico absoluto

```
find /home/username/public_html/images -name "articles.jpg"
```

Se precisar de pesquisar o nome do ficheiro, mas sem que a pesquisa seja CASE SENSITIVE, experimente:

```
find . -iname "Articles.jpg"
```

Veja a diferença com:

```
find . -iname "Articles.jpg" -exec ls -l {} \;
```

ou

```
find . -iname "Articles.jpg" -print0 | xargs -0 ls -l
```

## Usar o comando find no Linux com grep para pesquisar uma string ou texto

Imagine que precisamos de encontrar um pedaço de código num dos ficheiro PHP do Joomla, mas não sabemos em qual. Vamos pesquisar a palavra joomla em todos os ficheiros php:

```
find . -name "*.php" -print0 | xargs -0 grep -Hin "joomla"
```

ou

```
find . -name "*.php" -exec grep -Hin "joomla" {} \;
```

# Linux OS

E a ficheiros PHP alterados nas últimas 48 horas:

```
find . -name "*.php" -mtime -2 -exec grep -Hin --color=always "joomla" {} \;
```

Finalmente, vamos limitar o output aos ficheiros que incluem a string joomla, substituindo o grep -Hin com o grep -l:

```
find . -name "*.php" -print0 | xargs -0 grep -l "joomla"
```

ou

```
find . -name "*.php" -exec grep -l "joomla" {} \;
```

E se quisermos pesquisar em todos os ficheiros php e html?

```
find . \( -name "*.php" -o -name "*.html" \) -print0 | xargs -0 grep -Hin "joomla"
```

Se não quisermos usar o find para restringir a pesquisa a determinados tipo de ficheiros, a intervalos de data e tempo, a tamanhos de ficheiros, podemos usar apenas:

```
grep -Hirn --color=always joomla *
```

Autopsiando o grep -Hirn --color=always

-H imprime o nome dos ficheiros e respectivo caminho físico (no exemplo supra, poderíamos prescindir o H - imprime por defeito)

# Linux OS

-i pesquisa sem que seja CASE-SENSITIVE

-r faz uma pesquisa recursiva

-n imprime a linha no código onde se encontra a string pesquisada

--color=always usa cores no output de modo a tornar o output mais apresentável

Se quiser excluir uma diretoria da pesquisa:

```
--exclude-dir=dir
```

## Usar o comando find no Linux com base no Inode do ficheiro

Se estiver com problemas para apagar um ficheiro, dado que o nome do ficheiro é meio estranho, descubra o inode do ficheiro com o comando ls:

```
ls -il
```

e depois apague desta forma:

```
find -inum 117672808 -exec rm {} \;
```

## Usar o comando find no Linux para fintar o error argument list too long

Já tentou listar os ficheiros num diretório para retornar o erro:

Página 4 / 10

(c) 2024 Filipe Fonseca <suporte@ycorn.com.br> | 2024-05-20 22:51

URL: <https://ycorn.infoindex.php?action=artikel&cat=21&id=49&artlang=pt>

# Linux OS

```
bash: /bin/ls: Argument list too long
```

Experimente:

```
find . -type f -print0 | xargs -0 ls -l
```

ou

```
find . -maxdepth 1 -print0 | xargs -0 ls -l
```

ou ainda

```
find . -maxdepth 1 -type f -exec ls -l {} \; | less
```

Já encontrou a partição /tmp quase cheia e precisou de apagar os ficheiros, mas deparou com o erro

```
bash: /bin/rm: Argument list too long
```

## AVISO

Tenha especial cuidado com qualquer comando que apague ficheiros...

Vamos apagar os ficheiros, mas sem apagar quaisquer pastas e respectivos ficheiros e vamos deixar os links simbólicos (note que o find por defeito não segue os links simbólicos):

# Linux OS

```
find . -maxdepth 1 -type f -exec rm -f {} \;
```

O comando deve ser executado dentro da pasta /tmp

alternativa:

```
find . -maxdepth 1 -type f -print0 | xargs rm -f
```

Vamos agora usar a opção -delete do find:

```
find /tmp -maxdepth 1 -type f -delete
```

## Usar o comando find no Linux com base nas permissões dos ficheiros

Vamos agora pesquisar ficheiros com permissões 777 dentro dessa instalação do Joomla:

```
find . -perm 0777 -type f -exec ls -l {} \;
```

e pesquisar pastas com permissões 777:

```
find . -perm 0777 -type d -exec ls -l {} \;
```

## Usar o comando find no Linux com base no tamanho dos ficheiros

Já tentou pesquisar os 10 maiores ficheiros dentro duma instalação Joomla ou até dentro duma

# Linux OS

partição?

```
find . -type f -exec ls -s {} \; | sort -n -r | head -10
```

Neste exemplo, a listagem por tamanho dado pelo comando `ls -s` é muito importante. Veja como é diferente do uso do comando `ls` noutros exemplos, onde usamos `ls -l`. Experimente os 2 comandos:

```
ls -l
```

```
ls -s
```

O comando para pesquisar os 10 ficheiros mais pequenos é exactamente igual, com a diferença da parte do comando que ordena os ficheiros:

```
sort -n -r
```

aqui a ordem é descendente.

Para pesquisar os 10 ficheiros mais pequenos, use:

```
sort -n
```

Portanto, o comando completo é:

```
find . -type f -exec ls -s {} \; | sort -n | head -10
```

# Linux OS

E se precisarmos de pesquisar ficheiros maiores que 100 MB:

```
find . -size +100M -exec ls -s {} \;
```

Para pesquisar ficheiros vazios, podemos executar este comando:

```
find . -empty -exec ls -l {} \;
```

## Usar o comando find no Linux com base no tipo de ficheiros

Se quisermos pesquisar link simbólicos:

```
find . -type l -exec ls -l {} \;
```

Veja a diferença, se executar apenas:

```
find . -type l
```

Para pesquisarmos ficheiros escondidos:

```
find . -type f -name ".*"
```

Já experimentamos vários exemplos onde passamos ao comando find o tipo de ficheiro a pesquisar:

-type f (ficheiro)



# Linux OS

-type l (link simbólico)

-type d (diretório)

## Usar o comando find no Linux com base na data e no tempo dos ficheiros

Vamos agora pesquisar todos os ficheiros que foram acedidos nas últimas 24 horas:

```
find . -type f -atime -1 -exec ls -l {} \;
```

e todos os ficheiros que foram modificados nas últimas 24 horas:

```
find . -type f -mtime -1 -exec ls -l {} \;
```

Agora, em vez de nas últimas 24 horas, vamos pesquisar os ficheiros que foram modificados ontem:

```
find . -type f -mtime -1 -daystart -exec ls -l {} \;
```

Ao adicionar a opção -daystart, contamos as 24 horas a contar do início do dia de ontem.

Hoje, é Terça. Imagine que queremos saber que ficheiros foram modificados durante o fim de semana:

```
find . -type f -mtime 2 -mtime -3 -daystart -exec ls -l {} \;
```

# Linux OS

Vamos analisar as 3 opções de tempo:

-mtime = quando alteramos o conteúdo do ficheiro

-ctime = quando o inode associado ao ficheiro foi alterado - alterado sempre que alteramos o conteúdo, mas também quando mudamos o dono, as permissões do ficheiro, quando movemos para outra pasta

-atime = quando o ficheiro foi lido pela última vez

Um exercício que proponho que você faça é conjugar as várias opções que estou a exemplificar neste tutorial. Adapte estes exemplos às suas necessidades e comente sobre as suas descobertas. Partilhe connosco o que aprendeu. Ou, se já domina o comando find, partilhe connosco novos exemplos do comando find no linux e sugira alternativas aos exemplos aqui apresentados.

ID de solução Único: #1048

Autor: Martinho Novais

Atualização mais recente: 2014-01-31 12:49