

¿Cómo encuentro todos los archivos que contienen un texto específico en Linux?

Estoy tratando de encontrar una manera de escanear todo mi sistema Linux en busca de todos los archivos que contienen una cadena de texto específica. Solo para aclarar, estoy buscando texto dentro del archivo, no en el nombre del archivo.

Cuando estaba buscando cómo hacer esto, encontré esta solución dos veces:


```
find / -type f -exec grep -H 'texto-a-buscar-aqui' {} \;
```

Sin embargo, esto no funciona. Parece mostrar todos los archivos en el sistema.

¿Esto se parece a la forma correcta de hacerlo? Si no ¿cómo debería hacerlo? Esta capacidad de encontrar cadenas de texto en archivos sería extraordinariamente útil para algunos proyectos de programación que estoy haciendo.

linux text grep directory find

formulada hace 6 años

Nathan

7 Repuestas

Haz lo siguiente:

```
grep -rnw '/path/to/somewhere/' -e 'pattern'
```

- `-r` o `-R` es recursiva,
- `-n` es el número de línea y
- `-w` significa coincidir con la palabra completa.
- `-l` (L minúscula) se puede agregar para simplemente dar el nombre de archivo de los archivos coincidentes.

Juntos con estos, las `--exclude`, `--include`, `--exclude-dir` banderas podrian ser usadaas para una búsqueda eficiente.

- Esto solo buscará a través de aquellos archivos que tengan extensiones `.c` o `.h`:

```
grep --include=\*.{c,h} -rnw '/path/to/somewhere/' -e "pattern"
```

- Esto excluirá buscar todos los archivos terminados con extensión `.o`:

```
grep --exclude=*.o -rnw '/path/to/somewhere/' -e "pattern"
```

- Para los directorios es posible excluir un directorio/s particular/es a través del parametro `--exclude-dir` Por ejemplo, esto incluirá los dirs; `dir1/`, `dir2/` and all of them matching `*.dst/`:

```
grep --exclude-dir={dir1,dir2,*.dst} -rnw '/path/to/somewhere/' -e "pattern"
```

Esto me funciona muy bien, casi todos los mismos propósitos que los tuyos.

Por más opciones verifica `man grep` .



rakib_

Puedes usar `grep -iR` :

```
grep -Ril "texto-a-buscar-aqui" /
```

- `i` significa ignorar (es opcional en tu caso)
- `R` significa recursivo
- `l` significa "mostar el nombre del archivo, no el resultado en si mismo".
- `/` significa comenzar la raíz de tu máquina.



fedorqui

Puedes usar [ack](#). Es como [grep](#) para un código fuente. Puedes escanear tu sistema de archivos completo con esto.

Solo haz:

```
ack 'texto-a-buscar-aqui'
```

En tu directorio raíz.

Además puedes usar [regular expressions](#) que especifica el tipo de archivo etc.

ACTUALIZA

Acabo de descubri [El Buscador Silver](#) que es como ack pero 3-5x es mas rapido que este e incluso ignora los patrones desde un archivo `.gitignore` .



EarlOfEgo

Puedes usar:

```
grep -r "cadenas a buscar" /path/to/dir
```

La `r` significa recursivo y solo buscará el patrón especificado además de los subdirectorios. Esto te indicará el nombre del archivo y también imprimirá la línea en el archivo donde aparece la cadena.

O un comando similar al que estas probando (ejemplo:) para investigar en todos los archivos javascript (*.js):

```
find . -name '*.js' -exec grep -i 'cadena a buscar' {} \; -print
```

Esto imprimirá las líneas en los archivos donde aparece el texto, pero no imprime el nombre

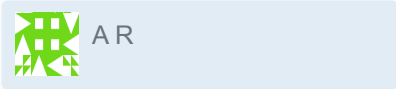
Además de este comando, también podemos escribir esto: `grep -rn "String to search" /path/to/directory/or/file` -r: búsqueda recursiva n: la linea de numeros sera mostrada en las coincidencias.



learner_19

Puedes usar esto:

```
grep -inr "Texto" folder/to/be/searched/
```



Lista de nombres de archivos que contienen un texto dado

Primero que todo, creoq ue has usado `-H` en vez de `-l` . Además puedes probar agregando el texto de citas seguido por `{}` `\` .

```
find / -type f -exec grep -l "texto-a-buscar-acqui" {} \;
```

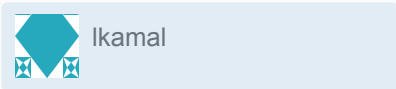
Ejemplo.

Digamos que estás buscando archivos que contengan el texto específico "Licencia de Apache" dentro de tu directorio. Mostrará resultados similares a los siguientes (la salida será diferente según el contenido de tu directorio).

```
bash-4.1$ find . -type f -exec grep -l "Licencia de Apache" {} \;
./net/java/jvnet-parent/5/jvnet-parent-5.pom
./commons-cli/commons-cli/1.3.1/commons-cli-1.3.1.pom
./io/swagger/swagger-project/1.5.10/swagger-project-1.5.10.pom
./io/netty/netty-transport/4.1.7.Final/netty-transport-4.1.7.Final.pom
./commons-codec/commons-codec/1.9/commons-codec-1.9.pom
./commons-io/commons-io/2.4/commons-io-2.4.pom
bash-4.1$
```

Eliminar la sensibilidad del caso

Incluso si no usas el caso como "texto" vs "TEXTO" puedes usar `-i` cambiar para ignorar el caso. Puedes leer detalles adicionales aquí. Espero que esto te ayude



grep (GNU or BSD)

Puedes usar la herramienta `grep` para buscar recursivamente la carpeta actual, como:

```
grep -r "class foo" .
```

También puedes usar la sintaxis globbing para buscar en archivos específicos como:

```
grep "class foo" **/*.c
```

Si tienes el error de que tu argumento es demasiado largo, considera limitar tu búsqueda o en vez de eso usa `find` tal como:

```
find . -name "*.php" -execdir grep -nH --color=auto foo {} ';' 
```

Alternativamente usa `ripgrep`.

ripgrep

Si estás trabajando en proyectos más grandes o archivos grandes, debes usar `ripgrep` como:

```
rg "class foo" .
```

Es mucha más rápida que cualquier otra herramienta como GNU/BSD `grep`, `ucg`, `ag`, `sift`, `ack`, `pt` o similar, ya que esta construido sobre [Rust's regex engine](#) el cual utiliza autómatas finitos, SIMD y optimizaciones literales agresivas para que la búsqueda sea muy rápida.

Esto soporta ignorar patrones especificados en archivos `.gitignore` por eso un simple patrón de archivo se puede hacer coincidir con múltiples patrones glob simultáneamente.

Puedes usar parámetros comunes tales como:

- `-i` - Búsqueda intensiva.
- `-I` - Ignora los archivos binarios
- `-w` - busca palabras completas (en oposición a coincidencias parciales de palabras)
- `-n` - Muestra la línea de tu coincidencia.
- `-C` / `--context` (e.g. `-C5`) - Aumenta el contexto, por eso verás el código circundante.
- `--color=auto` - Marca el texto que coincide.
- `-H` - Muestra el nombre de archivo donde el texto es encontrado.
- `-c` - Muestra el recuento de líneas coincidentes. Puede ser combinado con `-H`.



kenorb