



EBook Gratis

APRENDIZAJE

grep

Free unaffiliated eBook created from
Stack Overflow contributors.

#grep

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con grep.....	2
Observaciones.....	2
Referencias.....	2
Versiones.....	2
POSIX grep.....	2
Illumos / OpenSolaris grep.....	2
GNU grep.....	3
BSD grep / FreeGrep.....	4
Examples.....	5
Uso básico.....	5
Ignorar caso.....	6
Emparejar palabras completas.....	6
Buscar texto dentro de un directorio dado, recursivamente.....	7
Usando GNU grep.....	7
POSIX solución para buscar recursivamente.....	7
Imprime solo la parte coincidente de las líneas.....	7
Control de contexto de Grep.....	8
Capítulo 2: Control de línea de contexto.....	9
Observaciones.....	9
Examples.....	9
Imprimir líneas antes y / o después del patrón de coincidencia.....	9
Capítulo 3: Diferencia entre grep, egrep, fgrep, pgrep.....	12
Introducción.....	12
Sintaxis.....	12
Parámetros.....	12
Observaciones.....	13
Examples.....	14
grep con expresiones regulares básicas.....	14
egrep con expresiones regulares extendidas.....	14

fgrep sin expresiones regulares.....	14
pgrep con nombre de proceso.....	14
Capítulo 4: Expresiones regulares.....	15
Examples.....	15
Expresiones regulares.....	15
Mira atrás.....	15
Creditos.....	16

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [grep](#)

It is an unofficial and free grep ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official grep.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con grep

Observaciones

grep imprime líneas que contienen una coincidencia para un patrón dentro de los archivos.

grep puede usar [expresiones regulares](#) y tiene varias [opciones](#) para mejorar la calidad de los resultados.

Referencias

- [POSIX](#)
- [Página de manual de FreeBSD](#)
- [Página de manual de OpenBSD](#)
- [Manual en línea de GNU grep](#)
- [Página de manual de Illumos](#)

Versiones

POSIX grep

Versión	Fecha de lanzamiento
POSIX.2	1992-01-01
IEEE Std 1003.1-2001	2001-12-06
IEEE Std 1003.1, Edición 2004	2004-01-01
IEEE Std 1003.1, Edición 2013	2013-04-19
IEEE Std 1003.1, Edición 2016	2016-09-30

Illumos / OpenSolaris grep

Versión	Fecha de lanzamiento
2005-06-14	2005-06-14

Versión	Fecha de lanzamiento
2005-09-06	2005-09-06
2012-03-30	2012-03-30
2012-09-17	2012-09-17
2013-05-14	2013-05-14

GNU grep

Versión	Fecha de lanzamiento
2.0	1996-10-01
2.2	1998-04-27
2.3	1999-02-14
2.4.1	2000-03-01
2.4.2	2000-03-09
2.4	1999-12-03
2.5.1	2004-10-29
2.5.1a	2004-11-19
2.5.3	2007-08-02
2.5.4	2009-02-09
2.5	2002-03-13
2.6.1	2010-03-25
2.6.2	2010-03-29
2.6.3	2010-04-02
2.6	2010-03-23
2.7	2010-09-20
2.8	2011-05-13
2.9	2011-06-21

Versión	Fecha de lanzamiento
2,10	2011-11-16
2,11	2012-03-02
2.12	2012-04-23
2,13	2012-07-04
2,14	2012-08-20
2.15	2013-10-26
2.16	2014-01-01
2,17	2014-02-17
2.18	2014-02-20
2.19	2014-05-22
2.20	2014-06-03
2.21	2014-11-23
2,22	2015-11-01
2.23	2016-02-04
2.24	2016-03-10
2,25	2016-04-21
2,26	2016-10-02
2.27	2016-12-06
2,28	2017-02-06

BSD grep / FreeGrep

Versión	Fecha de lanzamiento
OpenBSD 3.0	2001-12-01
OpenBSD 3.4	2003-11-01
OpenBSD 3.5	2004-05-01

Versión	Fecha de lanzamiento
OpenBSD 3.6	2004-11-01
OpenBSD 3.7	2005-05-19
OpenBSD 3.8	2005-11-01
OpenBSD 3.9	2006-05-01
OpenBSD 4.0	2006-11-01
OpenBSD 4.1	2007-05-01
OpenBSD 4.3	2008-05-01
OpenBSD 4.8	2010-11-01
OpenBSD 5.0	2011-11-01
OpenBSD 5.3	2013-05-01
OpenBSD 5.7	2015-05-01
OpenBSD 5.8	2015-10-18
OpenBSD 5.9	2016-03-29
NetBSD 2.0	2004-12-09
NetBSD 4.0	2007-12-19
NetBSD 6.0	2012-10-17
NetBSD 7.0	2015-09-25
FreeBSD 9.0	2012-01-02
FreeBSD 10.0	2014-01-16

Examples

Uso básico

Ejecutando el comando:

```
grep sam someFile.txt
```

Cuando `someFile.txt` contiene:


```
fred 14 m foo
sam 68 m bar
christina 83 f baz
bob 22 m qux
Sam 41 m quux
```

Producirá esta salida:

```
sam 68 m bar
```

Ignorar caso

Dada una `sample` archivo:

```
hello
Hello
HELLO_there
```

Un `grep` normal para "hola" devuelve:

```
$ grep "hello" sample
hello
```

Usar `-i` permite ignorar mayúsculas y minúsculas y hacer coincidir cualquier "hola"

```
$ grep -i "hello" sample
hello
Hello
HELLO_there
```

Emparejar palabras completas

Dada una `sample` archivo:

```
hello world
ahello here
hello_there
```

Un `grep` normal para "hola" devuelve:

```
$ grep hello sample
hello world
ahello here
hello_there
```

El uso de `-w` permite seleccionar aquellas líneas que contienen coincidencias que forman palabras completas:

```
$ grep -w hello sample
hello world
```

Buscar texto dentro de un directorio dado, recursivamente

Usando GNU grep

```
grep -r 'pattern' <directory path>
```

Para enumerar también números de línea de coincidencias use la opción `-n`

```
grep -rn 'pattern' <directory path>
```

Para buscar solo archivos con un patrón [global](#) particular

```
grep --include='*.txt' -r 'pattern' <directory path>
```

Excluir patrones de archivos o directorios

```
grep -R --exclude=*.log 'pattern' <directory path>
grep -R --exclude={*.log,*.class} 'pattern' <directory path>

grep -R --exclude-dir=tmp 'pattern' <directory path>
grep -R --exclude-dir={tmp,lib} 'pattern' <directory path>
```

Notas y otras opciones útiles.

- `<directory path>` se puede omitir si se busca en el directorio actual
- Las opciones `-R` siguen todos los enlaces simbólicos, a diferencia de `-r` que sigue enlaces simbólicos solo si están en la línea de comando
- `-l` para listar solo archivos coincidentes
- `-h` para suprimir el prefijo de nombre de archivo
- `--color=auto` para resaltar patrones combinados
- `-m <num>` para especificar el número máximo de coincidencias para cada entrada de archivo

POSIX solución para buscar recursivamente

```
find <directory path> -type f -exec grep -l 'pattern' {} +
```

- Se pueden usar opciones como `-n`, `-l`, etc. según sea necesario
- Si `{}` + no es compatible, use `{}` \; en lugar
- Consulte la documentación de [encontrar](#) para obtener más ayuda sobre el comando de `find` como cómo incluir / excluir tipos de archivos, directorios, etc.

Imprime solo la parte coincidente de las líneas

```
echo "Prints only the matching part of the lines" | grep -o "matching"
```

```
# prints matching
```

Control de contexto de Grep

Dada una muestra de archivo llamada lista de películas.

```
Troy  
Gladiator  
Robin Hood  
King Arthur  
BraveHeart  
The Last Samurai
```

Grep normal devuelve

```
grep "Gladiator" movieslist  
Gladiator
```

Ahora, usando grep para imprimir las líneas de abajo o arriba del archivo.

Para imprimir la línea de abajo

```
grep -A 1 Gladiator movieslist  
Gladiator  
Robin Hood
```

Para imprimir la línea anterior

```
grep -B 1 Gladiator movieslist  
Troy  
Gladiator
```

Para imprimir ambos

```
grep -C 1 Gladiator movieslist  
Troy  
Gladiator  
Robin Hood
```

Lea Empezando con grep en línea: <https://riptutorial.com/es/grep/topic/2198/empezando-con-grep>

Capítulo 2: Control de línea de contexto

Observaciones

-A opciones -A , -B y -C no están disponibles en POSIX (consulte las [especificaciones POSIX para grep](#)).

Examples

Imprimir líneas antes y / o después del patrón de coincidencia

Por lo general, `grep` imprime solo líneas coincidentes. En el siguiente ejemplo, `seq 9` genera una lista de números del 1 al 9, uno por línea, y `grep` imprime una sola línea coincidente:

```
seq 9 | grep 5
# 5
```

La opción `-C n` (o `--context=n` en forma larga) imprime `n` líneas antes y después de cada línea coincidente, además de la línea correspondiente:

```
seq 9 | grep -C 2 '5'
# 3
# 4
# 5
# 6
# 7
```

Naturalmente, se imprimirán menos de `n` líneas si se llega al final del archivo o al principio del archivo.

Si queremos imprimir líneas solo antes o solo después, pero no ambas, podemos usar `-B n` (`--before-context=n`) o `-A n` (`--after-context=n`):

```
seq 9 | grep -B 2 '5'
# 3
# 4
# 5

seq 9 | grep -A 2 '5'
# 5
# 6
# 7
```

Tenga en cuenta que estas opciones no están disponibles en POSIX (consulte las [especificaciones de POSIX para grep](#)).

Si los contextos de dos o más líneas coincidentes se superponen, entonces todas las líneas se imprimen juntas como un gran contexto. En el siguiente ejemplo, `5` es parte del contexto de `3` y `7`:

```
seq 9 | grep -E --context=2 '3|7'  
# 1  
# 2  
# 3  
# 4  
# 5  
# 6  
# 7  
# 8  
# 9
```

Sin embargo, si los contextos no se superponen, se imprimen con una línea separadora de grupo. Por defecto esto es doble guión (--):

```
seq 9 | grep -E --context=2 '2|8'  
# 1  
# 2  
# 3  
# 4  
# --  
# 6  
# 7  
# 8  
# 9
```

Podemos establecer una línea separadora de grupo diferente usando la `--group-separator=SEP`, o suprimir esta línea completamente usando la `--no-group-separator`:

```
seq 9 | grep -E --context=0 --group-separator='****' '2|8'  
# 2  
# ****  
# 8  
  
seq 9 | grep -E --context=0 --group-separator='' '2|8'  
# 2  
#  
# 8  
  
seq 9 | grep -E --context=0 --no-group-separator '2|8'  
# 2  
# 8
```

Finalmente, si elegimos la opción `-v` para imprimir líneas no coincidentes, entonces se proporciona el contexto alrededor de esas líneas:

```
seq 9 | grep -E -v '1|3|4|5|6|7|9'  
# 2  
# --  
# 8  
  
seq 9 | grep -E -v -C 1 '1|3|4|5|6|7|9'  
# 1  
# 2  
# 3  
# --  
# 7
```

8

9

Lea Control de linea de contexto en línea: <https://riptutorial.com/es/grep/topic/4152/control-de-linea-de-contexto>

Capítulo 3: Diferencia entre grep, egrep, fgrep, pgrep.

Introducción

grep , **egrep** , **fgrep** , **rgrep** , **pgrep** : son comandos en sistemas operativos similares a Unix que imprimen líneas que coinciden con un patrón. El **grep** busca en el *archivo de entrada* con *nombre* las líneas que contienen una coincidencia con el *PATRÓN* dado. Por defecto, imprime las líneas correspondientes. Además, los programas de variante **egrep** , **fgrep** y **rgrep** son los mismos que **grep -E** , **grep -F** y **grep -r** , respectivamente. Estas variantes están en desuso, pero se proporcionan para compatibilidad con versiones anteriores.

Sintaxis

- `grep [OPTIONS] PATTERN [FILE...]`
- `grep [OPTIONS] [-e PATTERN]... [-f FILE]... [FILE...]`

Parámetros

Símbolo	Detalles Expresiones Regulares Básicas (BRE)
<code>^</code>	el circunflejo se utiliza para hacer coincidir el comienzo de una línea.
<code>\$</code>	utilizado para coincidir con el final de una línea.
<code>.</code>	coincide con cualquier carácter excepto una nueva línea.
<code>[]</code>	coincide con un solo carácter dentro de los paréntesis. Si hay un <code>^</code> interior, coincidiría con cualquier otra cosa que no sean los caracteres en el soporte.
<code>\</code>	antes de que cualquiera de los caracteres no alfanuméricos los cita.
<code>*</code>	el símbolo coincide con el carácter o subexpresión anterior cero, una o más veces.
<code>\1</code>	Las referencias inversas 1-9 coinciden con el texto exacto por el grupo correspondiente.
<code>\{m,n\}</code>	coincide con los elementos anteriores al menos <i>m</i> y no más de <i>n</i> veces.
<code> </code>	<code>foo bar</code> coincide con <code>foo</code> o <code>bar</code> .
<code>\?</code>	abreviatura de <code>{0,1}</code>

Símbolo	Detalles Expresiones Regulares Básicas (BRE)
\+	(abreviatura de {1,}) coincide con el carácter o subexpresión precedente como máximo 1 vez, o al menos 1 vez respectivamente.
\n	coincide con una nueva línea, \t coincide con una pestaña, etc.
\w	coincide con cualquier constituyente de la palabra y \W coincide con cualquier carácter que no sea un constituyente de la palabra.
\<\>	coincide con la cadena vacía solo al principio o al final de una palabra
\b	coincide ya sea con \ B coincide con donde \ b no lo hace.
Símbolo	Detalles de expresiones regulares extendidas (ERE)
^	partido solo al principio
\$	coincide solo al final de una línea.
.	coincide con cualquier carácter (o cualquier carácter excepto una nueva línea).
[...]	coincide con cualquiera de los caracteres enumerados dentro de los paréntesis (conjunto de caracteres). Agregue un ^ y los rangos iniciales funcionan como en BRE (vea arriba).
(...)	Grupo sintáctico, para usar con reemplazos * o \ DIGIT.
\	para la alternancia: foo bar coincide con foo o bar.
*	coincide con el carácter o subexpresión precedente varias veces: 0, 1 o más veces
+	coincide con 1 o más veces el carácter anterior.
?	coincide con los caracteres anteriores 0 o 1 veces.
\	La barra invertida cita el siguiente carácter si no es alfanumérico.
{m, n}	coincide con el carácter o subexpresión anterior entre m y n veces (falta en algunas implementaciones); n o m pueden omitirse, y {m} significa exactamente m

Observaciones

fgrep significa "Impresión de expresiones regulares globales de cadena fija". **fgrep** es lo mismo que **grep -F**. Este comando es un **grep** más rápido y se comporta como **grep** pero NO reconoce ningún meta-carácter de expresión regular como algo especial. La búsqueda se completará más rápido porque solo procesa una cadena simple en lugar de un patrón complejo.

pgrep es un acrónimo que significa "Impresión de expresiones regulares globales de ID de proceso". `pgrep` revisa los procesos que se están ejecutando actualmente y enumera los identificadores de proceso que coinciden con los criterios de selección para la salida estándar. `pgrep` es útil cuando lo único que quiere saber es el número de identificación de proceso de un proceso.

grep	egrep (grep -E)	fgrep (grep -F)	pgrep
Expresiones Regulares Básicas (BRE)	Expresiones regulares extendidas (ERE)	Busca solo cadenas	Búsquedas por nombre

Para más información y referencia use algunos de los siguientes enlaces:

[¿Cuál es la diferencia entre grep, egrep y fgrep? Unix y Linux StackExchange](#)

[¿Por qué mi expresión regular funciona en X pero no en Y? Unix y Linux StackExchange](#)

[¿Cuál es la diferencia entre grep, pgrep, egrep, fgrep? Superusuario](#)

Examples

grep con expresiones regulares basicas

```
$ grep root /etc/passwd
root:x:0:0:root:/root:/bin/bash
operator:x:11:0:operator:/root:/sbin/nologin
```

egrep con expresiones regulares extendidas

```
$ egrep '^(0|1)+ [a-zA-Z]+$' searchfile.txt
011 AaBBS
```

fgrep sin expresiones regulares

```
$ fgrep "." .bashrc
# will match lines with a dot.
```

pgrep con nombre de proceso

```
$ pgrep python
1299
```

Lea Diferencia entre `grep`, `egrep`, `fgrep`, `pgrep`. en línea:

<https://riptutorial.com/es/grep/topic/8936/diferencia-entre-grep--egrep--fgrep--pgrep->

Capítulo 4: Expresiones regulares

Examples

Expresiones regulares

El patrón de búsqueda también puede ser una expresión regular. Corriendo:

```
grep '^[A-Z]' someFile.txt
```

Cuando `someFile.txt` contiene:

```
fred 14 m foo
sam 68 m bar
christina 83 f baz
bob 22 m qux
Sam 41 m quux
```

Producirá la salida:

```
Sam 41 m quux
```

ya que esta es la única línea en `someFile.txt` comienza con una letra mayúscula.

Mira atrás

Dado el siguiente archivo:

```
hello how are you
i am fine
let's go, you!
let's go, baby!
```

`grep` with [look-behind](#) permite imprimir solo algunas partes:

```
$ grep -Po "(?<=let's go, ).*" file
you!
baby!
```

En este caso, coincide con lo que ocurre después de "vámonos".

Lea [Expresiones regulares en línea](https://riptutorial.com/es/grep/topic/4183/expresiones-regulares): <https://riptutorial.com/es/grep/topic/4183/expresiones-regulares>

Creditos

S. No	Capítulos	Contributors
1	Empezando con grep	Batsu , Benjamin W. , Community , David Pärsson , depperm , dingalapadum , fedorqui , Jerry Jeremiah , kdhp , mszymborski , Stuxnet78 , Sundeep , UNagaswamy
2	Control de linea de contexto	Benjamin W. , fedorqui , ghostarbeiter
3	Diferencia entre grep, egrep, fgrep, pgrep.	Bor
4	Expresiones regulares	David Pärsson , dingalapadum , fedorqui