

EXPRESSIONS REGULARS AMB grep

NOTA1: | () { } + ? necessiten \ per treballar com a metacaràcters (Línia 317 man grep mida consola per defecte).
Per evitar aquest problema, hem d'utilitzar grep -E.

NOTA 2: ^ [] . * \ no necessiten \ per treballar com a metacaràcters.

1-grep 'Jan' messages.1

Qualsevol línia que tingui la cadena Jan del fitxer messages.1 .

2-grep '^Jan'

Qualsevol línia que comenci per la cadena Jan del fitxer messages.1 .

3-grep '^Jan 22' messages.1

grep '^Jan\ 22' messages.1

Qualsevol línia que comenci amb Jan 24 del fitxer messages.1. L'espai en blanc no és un metacaràcter de manera que \ no té efecte.

4-grep '^Jan 22 12:14' messages.1

Qualsevol línia que comenci amb Jan 24 12:14 del fitxer messages.1 .

5-grep '^Jan 22 12:14:08' messages.1

Qualsevol línia que comenci amb Jan 24 12:14:08 del fitxer messages.1 .

6-grep 'seconds\$' messages.1

Qualsevol línia que finalitzi amb seconds del fitxer messages.1 .

7-grep '[Ss]econds\$' messages.1

Qualsevol línia que finalitzi amb seconds o Seconds del fitxer messages.1 .

8-grep '^Dec 23.*seconds\$' messages.1

Qualsevol línia que comenci amb Dec 23, tingui qualsevol combinació de caràcters repetidament i finalitzi amb seconds del fitxer messages.1 .

9- grep '^Dec 23.*[sS]econds\$' messages.1

Qualsevol línia que comenci amb Jan 24, tingui qualsevol combinació de caràcters repetidament i finalitzi amb seconds o Seconds del fitxer messages.1

10- grep -E '(Dec | Jan)' messages.1

Qualsevol línia que tingui la paraula Dec o Jan del fitxer messages.1

11- grep -E '(Apr.*2016)' messages.1

Qualsevol línia que tingui la paraula Apr i 2016 del fitxer messages.1

12- grep -E '(Apr 15.*2016)' messages.1

Qualsevol línia que tingui Apr 15 i 2016 del fitxer messages.1

13- grep -E '(Apr 15 | May 15)' messages.1

Qualsevol línia que tingui Apr 15 o May 15 del fitxer messages.1

14- grep -E '^ (Dec | Jan)' messages.1

Qualsevol línia que comenci amb Dec o Jan del fitxer messages.1 .

15- grep -E '^ (Dec | Jan) 22.*0\$' messages.1

Qualsevol línia que comenci per Dec o Jan, tingui després “ 22”, qualsevol combinació de lletres i finalitzi amb 0 del fitxer messages.1 .

16- grep '^[^#]' /etc/vsftpd.conf

Qualsevol línia que no comenci per # del fitxer /etc/vsftpd.conf .

17- grep '^[^#].*NOS' /etc/vsftpd.conf

Qualsevol línia que no comenci per # del fitxer /etc/vsftpd.conf i finalitzin amb NO, amb qualsevol combinació de lletres pel mig.

18- grep . /etc/host

Filtra i no mostra tots les línies en blanc d'un fitxer

- 19- **grep -E '1{1,}' /etc/hosts**
 Qualsevol línia que tinguin el número 1 repetit, de manera seguida, una o més vegades dins del fitxer /etc/hosts. És equivalent a **grep -E '1+' /etc/hosts** .
- 20- **grep -E 'f{2,}' /etc/hosts**
 Qualsevol línia que tinguin el caràcter f repetit, de manera seguida, 2 o més vegades dins del fitxer /etc/hosts .
- 21- **grep -E 'f{2}' /etc/hosts**
 Qualsevol línia que tinguin el caràcter f repetit, de manera seguida, 2 o més vegades dins del fitxer /etc/hosts .
- 22- **grep -E 'f{3,}' /etc/hosts**
 Qualsevol línia que tinguin el caràcter f repetit, de manera seguida, 3 o més vegades dins del fitxer /etc/hosts .
- 23- **grep -E 'f{5,6}' /proc/iomem**
 Qualsevol línia que tinguin el caràcter f repetit, de manera seguida, entre 5 i 6 vegades dins del fitxer /proc/iomem. En el cas que estigués 7 o més vegades, repetit de manera seguida, també podria sortir.
- 24 - **grep '[0-9]' prova**
 Qualsevol línia que tingui un número del 0 al 9 del fitxer prova.
- 25- **grep '^ [0-9]' prova**
 Qualsevol línia comenci per un número del 0 al 9 del fitxer prova.
- 26- **grep '^[:lower:]' boot.msg**
 Qualsevol línia que comenci per minúscules del fitxer prova.
- 27- **grep '^ [a-k]' boot.msg**
 Qualsevol línia que comenci per minúscules des de la lletra a fins la lletra k del fitxer prova.
- 28- **grep '[k-l]' boot.msg**
 Qualsevol línia que tingui les minúscules des de la lletra k fins la lletra l del fitxer prova.
- 29- **grep '^[:upper:]' prova**
 Qualsevol línia que tingui les majúscules des de la lletra k fins la lletra l del fitxer prova.
- 30- **grep '^ [A-D]' prova**
 Qualsevol línia que comenci per majúscules des de la lletra A fins la lletra D del fitxer prova.
- 31- **grep '^ [A-Df-g]' prova**
 Qualsevol línia que comenci per majúscules des de la lletra A fins la lletra C o minúscules de la f a la g del fitxer prova.
- 32- **grep '^ [A-Df-g3-5]' prova**
 Qualsevol línia que comenci per majúscules des de la lletra A fins la lletra C, o minúscules de la f a la g, o números del 3 al 5, del fitxer prova.
- 33- **grep '[:digit:]' prova**
 Qualsevol línia amb un dígit del fitxer prova.
- 34- **grep '\W' prova**
 Qualsevol línia que tingui un caràcter no alfabètic o numèric, com per exemple *,?, ,;:, &,%|,[,],#,(,/....., del fitxer prova.
- 35- **grep '127' prova** ==> Qualsevol línia que tingui la cadena '127' dins del fitxer prova.
grep '127.' prova ==> Qualsevol línia que tingui la cadena '127' i després un caràcter, dins del fitxer prova. Aquí no hi ha cap indicació numèrica. Si després poden haver més caràcters o no, és quèlcom que no està especificat.
grep -E '127.?' prova ==> Qualsevol línia que tingui la cadena '127' i després un caràcter 0 o més vegades del fitxer prova.
grep '127.\?' prova ==> Qualsevol línia que tingui la cadena '127' i després un caràcter 0 o més vegades del fitxer prova.
grep '127.?' prova ==> Qualsevol línia que tingui la cadena '127', després un caràcter, i després el caràcter ?, del fitxer prova.
grep -E '127.+ ' prova ==> Qualsevol línia que tingui la cadena '127' i després un caràcter 1 o més vegades del fitxer prova.

grep '127.\+' prova ==> Qualsevol línia que tingui la cadena '127' i després un caràcter 1 o més vegades del fitxer prova.

grep '127.+ ' prova ==> Qualsevol línia que tingui la cadena '127', després un caràcter, i després el caràcter +, del fitxer prova.

grep '127.*' prova ==> Qualsevol línia que tingui la cadena '127' i després un caràcter 0 o més vegades, del fitxer prova.

grep -E '127.{2}' prova ==> Qualsevol línia que tingui la cadena '127' i després un caràcter 2 o més vegades, del fitxer prova.

grep -E '127.{4,6}' prova ==> Qualsevol línia que tingui la cadena '127' i després un caràcter entre 4 i 6 vegades, del fitxer prova.

36- **grep 'pastanaga' prova** ==> Qualsevol línia que tingui la cadena 'pastanaga' dins del fitxer prova.

grep '\bpastanaga\b' prova ==> Qualsevol línia que tingui la cadena 'pastanaga' separada (abans de la paraula i després, hi ha el caràcter buit), dins del fitxer prova.

grep '\bpastanaga' prova ==> Qualsevol línia que tingui la cadena 'pastanaga' separada pel principi de la cadena anterior(abans de la paraula hi ha el caràcter buit), dins del fitxer prova.

grep 'pastanaga\b' prova ==> Qualsevol línia que tingui la cadena 'pastanaga' separada pel final de la següent cadena (després de la paraula hi ha el caràcter buit), dins del fitxer prova.

AWK

awk [-f arxiu_programa] [-Fc] ['programa'] [arxiu]
programa = 'Patró { Acció }

1- **tail -n 5 messages.1 | awk '{print \$3}'**

Mostra el 3r camp de les r últimes línies del fitxer messages.1. No hi ha patró.

2- **tail -n 5 messages.1 | awk '{print \$3; print \$4}'**

Mostra el 3r i 4t camps de les r últimes línies del fitxer messages.1. No hi ha patró.

3- **awk 'length < 50 {print \$0}' messages.1**

Mostra senceres (camp \$0) totes les línies de menys de 50 caràcters (patró length>50) del fitxer messages.1.

4- **awk 'length > 50 {print \$3}' messages.1**

Mostra el 3r camp de totes les línies de més de 50 caràcters (patró length>50) del fitxer messages.1.

5- **awk 'BEGIN {test=0; count=0;} \$1 ~ /Hola/ { count +=1; test += \$2 + \$3 + \$4; print \$0; print test} END{print count;}' awk_prova**

Inicialitza a 0 les variables test=0 i count=0. Busca totes les línies del fitxer awk_prova en les quals el primer camp sigui Hola i porta el compte de les línies trobades dins de la variable count. Dins de la variable test porta la suma acumulada dels camps 2,3 i 4 d'aquestes línies. Per cada línia trobada, mostra el contingut de la línia i el nou valor de test.Finalment, mostra el valor de count, que serà, el número de línies que han complert el patró.

6- **awk 'BEGIN {count=0;} \$3 ~ /casa/ {count +=1;print toupper(\$0)} END{print count;}' awk_prova**

Inicialitza count a 0. Troba totes les línies del fitxer awk_prova per les quals el 3r camp val casa, Cada cop que en troba una línia, la mostra per pantalla amb tot els caràcter passats a majúscules. Compta les línies trobades i mostra el valor al final.

7- **awk 'BEGIN {count=1;} {if (index(\$0,"entrecot")) print count;count+=1;}' awk_prova**

Inicialitza count a 1. Troba totes les línies del fitxer awk_prova dins de les qual es troba la paraula entrecot, i cada cop que la troba, indica a quina línia.

8- **awk 'BEGIN {count=1;} {i=index(\$0,"entrecot");if (i!=0) print count,i;count+=1;}' awk_prova**

Inicialitza count a 1. Troba totes les línies del fitxer awk_prova dins de les qual es troba la paraula entrecot, i cada cop que la troba, indica a quina línia i a quina posició de la línia .

9- **awk 'BEGIN {count=0;} /^Nov 23/ {count+=1;} END {printf "%d Línies trobades\n",count;}' messages.1**

Troba totes les línies que comencen per Nov 23 del fitxer messages.1 (o sigui, podem utilitzar expressions regulars), les compta, i dóna el resultat un cop finalitzat el procés, des del patró END.

10- **awk '/^Oct 15.*Seconds\$/ {for (i=1;i<=NF;i++) print \$i ;}' messages.1**

Mostra cada camp en una línia diferent, de totes aquelles línies que comencen per Oct 15 i finalitzen en Seconds del fitxer messages.1. El valor NF en AWK té el número de camps d'una línia.

11- `awk '/^Oct 15.*Seconds$/ {for (i=1;i<=NF;i++) print $i ; printf "Número de camps=%d\n",NF;}' messages.1`
Mostra cada camp en una línia diferent, de totes aquelles línies que comencen per Oct 15 i finalitzen en Seconds del fitxer messages.1. També escriu per pantalla el número de camps trobats.

12- `awk '/Principi/,/Final/ {print $0}' awk_prova`
Mostra totes les línies entre Principi i Final del fitxer awl_prova.

WEBS AMB DOCUMENTACIÓ

Expr. Regulars

<http://www.regular-expressions.info/reference.html>

http://www.bdat.net/documentos/expresiones_regulares/

http://www.gnu.org/software/grep/doc/grep_12.html#SEC12 (Metacaràcters amb \ al davant)

AWK

<http://club.telepolis.com/jagar1/Unix/Awk.htm>

<http://www.gnu.org/software/gawk/manual/gawk.html>

http://people.cs.uu.nl/piet/docs/nawk/nawk_toc.html

EXPRESSIONS REGULARS ESPECIALS

Expressions regulars per comprovar si és una URL correctament escrita:

```
^(https?|ftp|file)://[-A-Za-z0-9\+\&@#/%?=\~_!:\.,]*[-A-Za-z0-9\+\&@#/%=\~_]
```

```
^(ht|f)tp(s?):\V\{0-9a-zA-Z\}([-.\w]*[0-9a-zA-Z])*(:(0-9)*)(\/?)([a-zA-Z0-9\-\.\!\?,\'\V\|+\&#\$_]*)?$
```

Expressió regular per comprovar adreça de correu electrònic correctament escrit:

```
^[A-Z0-9a-z._%+-]+@[A-Z0-9a-z.-]+\.[A-Za-z]{2,63}$
```

Expressió regular per comprovar si és un NIF correctament escrit:

```
[0-9]{8}[A-Z] NIF
```

Expressió regular per comprovar si IP o MAC escrites correctament:

```
IP --> ^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}$ (no comprova si el número està entre 0 i 255)
```

```
MAC --> ^([0-9A-Fa-f]{2}[:-]){5}([0-9A-Fa-f]{2})$
```

ORDRE IF AMB EXPRESSIONS REGULAR

```
#!/bin/bash
clear
regex='^[a-z] '
cadena1='Badae'
cadena2='aRgadae'
# =~ comprova si %cadena1 té un format que quadra amb l'expressió regular
if [[ $cadena1 =~ $regex ]]
then
    echo "$cadena1 comença amb una lletra minúscula"
else
    echo "$cadena1 no comença amb una lletra minúscula"
fi
# =~ comprova si %cadena2 té un format que quadra amb l'expressió regular
if [[ $cadena2 =~ $regex ]]
then
    echo "$cadena2 comença amb una lletra minúscula"
else
    echo "$cadena2 no comença amb una lletra minúscula"
fi
exit 0
```