Grep e le Espressioni Regolari

grep [*opzioni*] *pattern* [*nomefile*]

- Stampa le righe del file che corrispondono al pattern
- Il pattern è una *espressione regolare*
- Nel caso più semplice, il pattern può essere una stringa senza caratteri speciali:

```
grep a pippo.txt
```

stampa le righe di pippo.txt che contengono una a

grep [*opzioni*] *pattern* [*nomefile*]

- Se nomefile non è specificato, legge da standard input
- Questo consente la concatenazione in *pipe*

`ls -l | grep 2006`	elenca i file che sono stati modificati l'ultima volta nel 2006 (ma non solo ...)
`ls -l | grep rwx`	elenca i file per cui almeno una categoria di utenti ha tutti i permessi (ma non solo ...)

- Con opzione “-v”, stampa le righe che non corrispondono al pattern

`ls -l | grep -v doc`	elenca i file che non contengono “doc” nel nome

- Con “-c”, visualizza solo il numero di occorrenze della stringa nel file; “-i” case-Insensitive; “-n” numero di riga

grep [*opzioni*] *pattern* [*nomefile*]

```
lso:~>grep root /etc/passwd
root:x:0:0:root:/root:/bin/bash
operator:x:11:0:operator:/root:/sbin/nologin

lso:~>grep -n root /etc/passwd
1:root:x:0:0:root:/root:/bin/bash
12:operator:x:11:0:operator:/root:/sbin/nologin

lso:~>grep -c root /etc/passwd
2
```

grep [*opzioni*] *pattern* [*nomefile*]

```
lso:~>grep -v bash /etc/passwd |grep -v nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
news:x:9:13:news:/etc/news:

lso:~>grep -c account /etc/passwd
0
lso:~>grep -c -i account /etc/passwd
1
lso:~>grep -i account /etc/passwd
lso:x:501:501:LSO Account:/home/lso:/bin/bash
lso:~>grep -i account /etc/passwd |wc -l
   1
```

Basic Regular Expressions

- Una espressione regolare e' un *pattern che descrive un insieme di stringhe*
- *L'elemento atomico delle espressioni regolari e' il carattere*
  - *Un carattere e' una espressione regolare che descrive se stesso*
  - *L'espressione “a” descrive “l'insieme di stringhe {a}”*
- *La maggior parte dei caratteri sono “espressioni regolari”*
- *I Metacaratteri corrispondono ad operatori*
  - *Un metacarattere puo' essere utilizzato con il suo valore utilizzando il carattere di escape “\”*

Basic Regular Expressions

. (un punto)	qualunque carattere	(1)
exp*****	zero o più occorrenze di exp	(2)
^exp	exp all'inizio del rigo	(1)
exp**$**	exp alla fine del rigo	(1)
[a-z**]**	un carattere nell'intervallo specificato	
[^a-z**]**	un carattere fuori dall'intervallo	

Note: (1) è un carattere normale per Bash
(2) ha un significato diverso per Bash

Basic Regular Expressions

\<exp	exp all'inizio di una parola	(1)
exp**\>**	exp alla fine di una parola	(1)
exp{N}	exp compare N volte	(1)
exp{N,}	exp compare almeno N volte	(1)
exp{N,M}	exp almeno N volte e al più M	(1)
[[:CLASS:]]	un carattere in CLASS	(1)

Note: (1) è un carattere normale per Bash
(2) ha un significato diverso per Bash

Basic Regular Expressions

Le classi di caratteri POSIX:

[[:alpha:]]	I caratteri alfabetici
[[:alnum:]]	I caratteri alfanumerici
[[:digit:]]	Le cifre
[[:upper:]]	I caratteri alfabetici maiuscoli
[[:lower:]]	I caratteri alfabetici minuscoli

Esempi

a*b	zero o più “a” seguite da una “b”
a.*b	una “a” prima di una “b”
\<[[:upper:]]	una parola che inizia con lettera maiuscola
^d	la lettera “d” all'inizio del rigo
^a*$	un rigo vuoto o composto solo di “a”
^a.*b$	un rigo che inizia con “a” e finisce con “b”
\<.-	una parola con un trattino al secondo posto

Basic Regular Expressions

Molti comandi per l'elaborazione di testi di UNIX (ad esempio grep, ed, sed, ..) consentono la definizione di espressioni regolari, ossia di schemi per la ricerca di testo basati sull'impiego di metacaratteri:

- Generalmente, i metacaratteri usati da tali comandi non coincidono con i metacaratteri impiegati dalla shell per identificare i nomi dei file
- Molti caratteri che hanno un significato speciale nelle espressioni regolari hanno pure un significato speciale per la shell

- Attenzione a non confondere i metacaratteri di shell con quelli che non lo sono
- Utilizzare gli apici ' ' o i doppi apici " " per racchiudere le espressioni

Basic Regular Expressions

- La “concatenazione” di espressioni regolari e' una espressione regolare:
  - Le “stringhe” possono essere costruite dalla “concatenazione” dei caratteri.
  - Una stringa corrisponde (“match”) ad una concatenazione di stringhe se e' composta da due sottostringhe che corrispondono, rispettivamente, alle due espressioni regolari
  - “ab” corrisponde alla concatenazione di exp1=“a” ed exp2=“b”
- L'operatore “|” (es. exp3=exp1|exp2)
  - Una stringa corrisponde ad exp3 se esiste un match con exp1 o con exp2.

Extended Regular Expressions

exp**+**	una o più occorrenze di exp	(1)
exp**?**	zero o una occorrenza di exp	(2)
exp1 **|** exp2	exp1 oppure exp2	(2)
(exp **)**	equivalente a exp, serve a stabilire l'ordine di valutazione	

In **grep**, questi simboli vanno preceduti da "****" (backslash)
In **egrep** (*extended grep*), si usano direttamente

Note: (1) è un carattere normale per Bash
(2) ha un significato diverso per Bash

Esempi per grep

`[[:digit:]]\+`	una sequenza non vuota di cifre
`^a|b`	un rigo che inizia con a oppure
	contiene b (precedenza)
`^\(a|b\)`	un rigo che inizia con a oppure con b
`\(\(\.txt\)|\(\.doc\)\)\>`	
	una parola che termina con .txt o con .doc
	(in egrep, “`((\.txt)|(\.doc))\>`”)

Esempi per grep

```
lso:~>egrep '^r.*n$|^r.*37' /etc/passwd
rpm:x:37:37::/var/lib/rpm:/bin/bash
rpc:x:32:32:Portmapper RPC user:/:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
```

```
lso:~>grep '^r.*n$|^r.*37' /etc/passwd
lso:~>grep '^r.*n$\|^r.*37' /etc/passwd
rpm:x:37:37::/var/lib/rpm:/bin/bash
rpc:x:32:32:Portmapper RPC user:/:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
```

Esercizi

1. Elencare i file con permesso di esecuzione per il proprietario
2. Elencare le directory il cui nome inizia per maiuscola
3. Elencare i file con permesso di esecuzione oppure di scrittura per il gruppo di appartenenza

Riferimenti

Capitolo 4 di [Bash Guide for Beginners]