RIFERIMENTO RAPIDO GREP

Corrispondenza pattern, regex, ricerca ricorsiva, contesto, filtraggio

Utilizzo Base

Eseguire grep

```
grep "pattern" file.txt            # cerca nel file
grep "error" *.log                 # cerca in più file
grep "hello" file1.txt file2.txt   # lista file esplicita
cat file.txt | grep "pattern"      # input da pipe
dmesg | grep -i "usb"              # filtra output comandi
```

Flag Comuni

`-i`	Corrispondenza case-insensitive
`-v`	Inverti corrispondenza — stampa righe non corrispondenti
`-c`	Stampa il conteggio delle righe corrispondenti
`-n`	Mostra i numeri di riga
`-l`	Elenca solo i nomi dei file con corrispondenze
`-L`	Elenca i nomi dei file senza corrispondenze
`-w`	Corrisponde solo a parole intere
`-x`	Corrisponde solo a righe intere

Pattern Regex

Espressioni Regolari Base (BRE)

`.`	Qualsiasi singolo carattere
`*`	Zero o più dell'elemento precedente
`^`	Inizio riga
`$`	Fine riga
`[abc]`	Classe caratteri — uno qualsiasi tra a, b, c
`[^abc]`	Classe negata — qualsiasi tranne a, b, c
`[a-z]`	Intervallo — lettere minuscole
`\<, \>`	Confini di parola (GNU)
`\( \), \1`	Gruppo di cattura e back-reference

Esempi BRE

```
grep '^#' file.conf                # righe che iniziano con #
grep 'error$' file.log             # righe che terminano con error
grep '^$' file.txt                 # righe vuote
grep 'col[ou]r' file.txt           # corrisponde a color o colour
```

Regex Estesa

Espressioni Regolari Estese (ERE)

`+`	Uno o più dell'elemento precedente
`?`	Zero o uno dell'elemento precedente
`{n}`	Esattamente n ripetizioni
`{n,m}`	Tra n e m ripetizioni
`(a|b)`	Alternanza — corrisponde a a o b
`( )`	Raggruppamento (senza backslash)

Esempi ERE

```
grep -E '[0-9]{3}-[0-9]{4}' f     # pattern numero di telefono
grep -E '(error|warn|fatal)' f     # pattern multipli
grep -E '^[A-Z][a-z]+' f           # parole con iniziale maiuscola
grep -P '\d{1,3}\.\d{1,3}' f      # regex Perl: frammenti IP
```

Righe di Contesto

Esempi Contesto

```
grep -B 3 "error" app.log          # 3 righe prima della
corrispondenza
grep -A 5 "FAIL" test.log          # 5 righe dopo la
corrispondenza
grep -C 2 "crash" kern.log         # 2 righe prima e dopo
grep --group-separator="---" -C 1 "err" f  # separatore
personalizzato
```

Flag Contesto

`-B N`	Mostra N righe prima di ogni corrispondenza
`-A N`	Mostra N righe dopo ogni corrispondenza
`-C N`	Mostra N righe prima e dopo (contesto)
`--group-separator=str`	Separatore tra gruppi di corrispondenze (default `--`)
`--color=auto`	Evidenzia le corrispondenze nel terminale

Ricerca Ricorsiva

Esempi Ricorsivi

```
grep -r "TODO" .                   # ricorsivo dalla directory
corrente
grep -rn "FIXME" src/              # ricorsivo con numeri di riga
grep -r --include="*.py" "import" .  # solo file .py
grep -r --exclude="*.log" "error" .  # salta file .log
grep -r --exclude-dir=node_modules "require" .
```

Flag Ricorsivi

`-r / --recursive`	Cerca ricorsivamente nelle directory
`-R`	Come `-r` ma segue i link simbolici
`--include=glob`	Cerca solo nei file corrispondenti al glob
`--exclude=glob`	Salta i file corrispondenti al glob
`--exclude-dir=dir`	Salta le directory corrispondenti al nome
`--include-dir=dir`	Cerca solo nelle directory corrispondenti

Conteggio ed Elenco

Esempi Conteggio ed Elenco

```
grep -c "error" *.log              # conta corrispondenze per file
grep -l "TODO" src/*.py            # elenca file con TODO
grep -L "test" src/*.py            # file senza "test"
grep -o "http[^ ]*" page.html      # estrai solo le parti
corrispondenti
grep -c '' file.txt                # conta righe totali (come wc -
l)
```

Flag Output

`-c`	Stampa il conteggio delle righe corrispondenti per file
`-l`	Stampa solo i nomi dei file con corrispondenze
`-L`	Stampa solo i nomi dei file senza corrispondenze
`-o`	Stampa solo le parti corrispondenti delle righe
`-H / -h`	Mostra / nasconde il prefisso del nome file
`-Z`	Output delimitato da null (per xargs -0)

Corrispondenza Inversa

Inverti ed Escludi

```
grep -v "^#" config.conf           # rimuovi righe di commento
grep -v "^$" file.txt              # rimuovi righe vuote
grep -v -e "debug" -e "trace" app.log  # escludi due pattern
grep -v "pattern" f | grep "other" # catena: NON A, poi B
```

Strategie di Filtraggio

`-v`	Inverti corrispondenza — seleziona righe non corrispondenti
`-v con -e`	Escludi pattern multipli
`catena pipe`	Concatena chiamate grep per filtraggio complesso
`grep -v '^$' | grep -v '^#'`	Rimuovi righe vuote e commenti
`-v con -c`	Conta righe non corrispondenti

Pattern Multipli

Esempi Pattern Multipli

```
grep -e "error" -e "warning" app.log
grep -E "error|warning|fatal" app.log
grep -f patterns.txt file.txt      # pattern da file
grep -w -e "GET" -e "POST" access.log
```

Opzioni Pattern

`-e pattern`	Specifica un pattern (uso multiplo)
`-f file`	Legge pattern da file (uno per riga)
`-E 'a|b|c'`	Alternanza ERE per pattern multipli
`-F`	Stringhe fisse — nessuna regex, corrispondenza più veloce
`-G`	Regex base (modalità predefinita)
`-P`	Regex compatibile Perl (PCRE)

Performance

Consigli sulle Performance

`-F (fgrep)`	Modalità stringa fissa — più veloce per stringhe letterali
`LC_ALL=C grep`	Bypassa la localizzazione per un aumento 2-10x su dati ASCII
`--include/--exclude`	Riduci i file da cercare prima di aprirli
`-m N`	Ferma dopo N corrispondenze per file
`-q`	Modalità silenziosa — esce alla prima corrispondenza (per script)
`ripgrep (rg)`	Sostituto diretto; più veloce su grandi repository

Esempi di Performance

```
LC_ALL=C grep -F "exact string" huge.log
grep -r -m 1 "needle" /var/log/    # ferma dopo il primo
risultato
grep -rq "pattern" . && echo "found" # test booleano
grep -r --include="*.go" "func main" .
```

Pattern Comuni

One-Liner

```
grep -rn "TODO\|FIXME\|HACK" src/      # trova marcatori codice
grep -oP '(?<=")[^"]+(?=")' f          # estrai stringhe tra
virgolette
grep -E '^\s*$' f | wc -l              # conta righe vuote
grep -c '' *.py | sort -t: -k2 -rn     # ordina file per conteggio
righe
grep -rn --include="*.yaml" "password" . # audit per segreti
```

Ricette

`Indirizzi IP`	`grep -oE '[0-9]{1,3}(\.[0-9]{1,3}){3}'`
`Indirizzi email`	`grep -oE '[a-zA-Z0-9._%+-]+@[a-z.-]+'`
`URL`	`grep -oE 'https?://[^ ]+'`
`Righe tra marcatori`	`grep -A999 'START' f | grep -B999 'END'`
`Corrispondenze univoche`	`grep -oE 'pattern' f | sort -u`
`Conteggio per pattern`	`grep -c 'pat1' f; grep -c 'pat2' f`