

Referência Rápida grep

Correspondência de padrões, regex, busca recursiva, contexto, filtragem

Uso Básico

Executando grep

```
grep "pattern" file.txt # search in file
grep "error" *.log # search multiple files
grep "hello" file1.txt file2.txt # explicit file list
cat file.txt | grep "pattern" # pipe input
dmesg | grep -i "usb" # filter command output
```

Flags Comuns

- i** Correspondência sem diferença de maiúsculas/minúsculas
- v** Inverter correspondência — imprimir linhas não correspondentes
- c** Imprimir contagem de linhas correspondentes
- n** Mostrar números de linha
- l** Listar apenas nomes de arquivos com correspondências
- L** Listar nomes de arquivos sem correspondências
- w** Corresponder apenas palavras inteiras
- x** Corresponder apenas linhas inteiras

Padrões Regex

Expressões Regulares Básicas (BRE)

.	Qualquer caractere único
*	Zero ou mais do elemento anterior
^	Início de linha
\$	Fim de linha
[abc]	Classe de caractere — qualquer a, b ou c
[^abc]	Classe negada — qualquer coisa exceto a, b, c
[a-z]	Faixa — letras minúsculas
\<, \>	Limites de palavra (GNU)
\(\), \1	Grupo de captura e retroreferência

Exemplos BRE

```
grep '^#' file.conf # lines starting with #
grep 'error$' file.log # lines ending with error
grep '^$' file.txt # blank lines
grep 'col[ou]r' file.txt # match color or colour
```

Regex Estendida

Expressões Regulares Estendidas (ERE)

+	Um ou mais do elemento anterior
?	Zero ou um do elemento anterior
{n}	Exatamente n repetições
{n,m}	Entre n e m repetições
(a b)	Alternância — corresponder a ou b
()	Agrupamento (sem barra invertida necessária)

Exemplos ERE

```
grep -E '[0-9]{3}-[0-9]{4}' f # phone number pattern
grep -E '(error|warn|fatal)' f # multiple patterns
grep -E '^[A-Z][a-z]+' f # capitalized words
grep -P '\d{1,3}\.\d{1,3}' f # Perl regex: IP fragments
```

Linhas de Contexto

Exemplos de Contexto

```
grep -B 3 "error" app.log # 3 lines before match
grep -A 5 "FAIL" test.log # 5 lines after match
grep -C 2 "crash" kern.log # 2 lines before and after
grep --group-separator="---" -C 1 "err" f # custom separator
```

Flags de Contexto

-B N	Mostrar N linhas antes de cada correspondência
-A N	Mostrar N linhas após cada correspondência
-C N	Mostrar N linhas antes e depois (contexto)
--group-separator=str	Separador entre grupos de correspondência (padrão --)
--color=auto	Destacar correspondências no terminal

Busca Recursiva

Exemplos Recursivos

```
grep -r "TODO" . # recursive from current dir
grep -rn "FIXME" src/ # recursive with line numbers
grep -r --include="*.py" "import" . # only .py files
grep -r --exclude="*.log" "error" . # skip .log files
grep -r --exclude-dir=node_modules "require" .
```

Flags Recursivas

-r / --recursive	Buscar diretórios recursivamente
-R	Como -r mas segue links simbólicos
--include=glob	Buscar apenas arquivos que correspondem ao glob
--exclude=glob	Pular arquivos que correspondem ao glob
--exclude-dir=dir	Pular diretórios que correspondem ao nome
--include-dir=dir	Buscar apenas diretórios que correspondem ao nome

Contagem e Listagem

Exemplos de Contagem e Listagem

```
grep -c "error" *.log # count matches per file
grep -l "TODO" src/*.py # list files with TODOs
grep -L "test" src/*.py # files missing "test"
grep -o "http[ ]*" page.html # extract matching parts only
grep -c '' file.txt # count total lines (like wc -l)
```

Flags de Saída

-c	Imprimir contagem de linhas correspondentes por arquivo
-l	Imprimir apenas nomes de arquivos com correspondências
-L	Imprimir apenas nomes de arquivos sem correspondências
-o	Imprimir apenas as partes correspondentes das linhas
-H / -h	Mostrar / ocultar prefixo do nome de arquivo
-Z	Saída delimitada por null (para xargs -0)

Correspondência Inversa

Inverter e Excluir

```
grep -v "^#" config.conf # remove comment lines
grep -v "^$" file.txt # remove blank lines
grep -v -e "debug" -e "trace" app.log # exclude two patterns
grep -v "pattern" f | grep "other" # chain: NOT A, then B
```

Estratégias de Filtragem

-v	Inverter correspondência — selecionar linhas não correspondentes
-v with -e pipe chain	Excluir múltiplos padrões Encadear chamadas grep para filtragem complexa
grep -v '^\$' grep -v '^#'	Remover linhas em branco e comentários
-v with -c	Contar linhas não correspondentes

Múltiplos Padrões

Exemplos de Múltiplos Padrões

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

Opções de Padrão

-e pattern	Especificar um padrão (usar múltiplas vezes)
-f file	Ler padrões de arquivo (um por linha)
-E 'a b c'	Alternância ERE para múltiplos padrões
-F	Strings fixas — sem regex, correspondência mais rápida
-G	Regex básica (modo padrão)
-P	Regex compatível com Perl (PCRE)

Performance

Dicas de Performance

-F (fgrep)	Modo de string fixa — mais rápido para strings literais
LC_ALL=C grep	Ignorar locale para aumento de 2-10x em dados ASCII
--include/--exclude -m N	Reduzir arquivos pesquisados antes de abrir Parar após N correspondências por arquivo
-q	Modo silencioso — sair na primeira correspondência (para scripts)
ripgrep (rg)	Substituto drop-in; mais rápido em repositórios grandes

Exemplos de Performance

```
LC_ALL=C grep -F "exact string" huge.log
grep -r -m 1 "needle" /var/log/ # stop after first hit
grep -rq "pattern" . && echo "found" # boolean test
grep -r --include="*.go" "func main" .
```

Padrões Comuns

One-Liners

```
grep -rn "TODO|FIXME|HACK" src/ # find code markers
grep -oP '(?<=)[^"]+(?=")' f # extract quoted strings
grep -E '^s*$' f | wc -l # count blank lines
grep -c '' *.py | sort -t: -k2 -rn # sort files by line count
grep -rn --include="*.yaml" "password" . # audit for secrets
```

Receitas

IP addresses	grep -oE '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}'
Email addresses	grep -oE '[a-zA-Z0-9._%+-]@[a-z.-]+'
URLs	grep -oE 'https://[^\]+'
Lines between markers	grep -A999 'START' f grep -B999 'END'
Unique matches	grep -oE 'pattern' f sort -u
Count per pattern	grep -c 'pat1' f; grep -c 'pat2' f