# grep命令详解

grep （global search regular expression(RE) and print out the line，全面搜索正则表达式并把行打印出来）是一种强大的文本搜索工具，它能使用正则表达式搜索文本，并把匹配的行打印出来。用于过滤/搜索的特定字符。可使用正则表达式能多种命令配合使用，使用上十分灵活。

# 语法

grep [option] files

# option:

|  |  |
| --- | --- |
| -a --text | 不要忽略二进制数据。 |
| -A <显示行数> --after-context=<显示行数> | 除了显示符合范本样式的那一行之外，并显示该行之后的内容。 |
| -b --byte-offset | 在显示符合范本样式的那一行之外，并显示该行之前的内容。 |
| -B<显示行数> --before-context=<显示行数> | 除了显示符合样式的那一行之外，并显示该行之前的内容。 |
| -c --count | 计算符合范本样式的列数。 |
| -C<显示行数> --context=<显示行数>或-<显示行数> | 除了显示符合范本样式的那一列之外，并显示该列之前后的内容。 |
| -d<进行动作> --directories=<动作> | 当指定要查找的是目录而非文件时，必须使用这项参数，否则grep命令将回报信息并停止动作。 |
| -e<范本样式> --regexp=<范本样式> | 指定字符串作为查找文件内容的范本样式。 |
| -E --extended-regexp | 将范本样式为延伸的普通表示法来使用，意味着使用能使用扩展正则表达式。 |
| -f<范本文件> --file=<规则文件> | 指定范本文件，其内容有一个或多个范本样式，让grep查找符合范本条件的文件内容，格式为每一列的范本样式。 |
| -F --fixed-regexp | 将范本样式视为固定字符串的列表。 |
| -G --basic-regexp | 将范本样式视为普通的表示法来使用。 |
| -h --no-filename | 在显示符合范本样式的那一列之前，不标示该列所属的文件名称。 |
| -H --with-filename | 在显示符合范本样式的那一列之前，标示该列的文件名称。 |
| -i --ignore-case | 忽略字符大小写的差别。 |
| -l --file-with-matches | 列出文件内容符合指定的范本样式的文件名称。 |
| -L --files-without-match | 列出文件内容不符合指定的范本样式的文件名称。 |
| -n --line-number | 在显示符合范本样式的那一列之前，标示出该列的编号。 |
| -q --quiet或--silent | 不显示任何信息。 |
| -R/-r --recursive | 此参数的效果和指定“-d recurse”参数相同。 |
| -s --no-messages | 不显示错误信息。 |
| -v --revert-match | 反转查找即查找除匹配行外的其他行 |
| -V --version | 显示版本信息。 |
| -w --word-regexp | 只显示全字符合的列。 |
| -x --line-regexp | 只显示全列符合的列。 |
| -y | 此参数效果跟“-i”相同。 |
| -o | 只输出文件中匹配到的部分。 |
| -m <num> --max-count=<num> | 找到num行结果后停止查找，用来限制匹配行数 |

# 正则表达式：

|  |  |
| --- | --- |
| ^ | 锚定行的开始如：'^grep'匹配所有以grep开头的行。 |
| $ | 锚定行的结束如：'grep$'匹配所有以grep结尾的行。 |
| . | 匹配一个非换行符的字符如：'gr.p'匹配gr后接一个任意字符，然后是p。 |
| \* | 匹配零个或多个先前字符如：'\*grep'匹配所有一个或多个空格后紧跟grep的行。 |
| .\* | 一起用代表任意字符。 |
| [] | 匹配一个指定范围内的字符，如'[Gg]rep'匹配Grep和grep。 |
| [^] | 匹配一个不在指定范围内的字符，如：'[^A-FH-Z]rep'匹配不包含A-R和T-Z的一个字母开头，紧跟rep的行。 |
| \(..\) | 标记匹配字符，如'\(love\)'，love被标记为1。 |
| \< | 锚定单词的开始，如:'\<grep'匹配包含以grep开头的单词的行。 |
| \> | 锚定单词的结束，如'grep\>'匹配包含以grep结尾的单词的行。 |
| x\{m\} | 重复字符x，m次，如：'0\{5\}'匹配包含5个o的行。 |
| x\{m,\} | 重复字符x,至少m次，如：'o\{5,\}'匹配至少有5个o的行。 |
| x\{m,n\} | 重复字符x，至少m次，不多于n次，如：'o\{5,10\}'匹配5--10个o的行。 |
| \w | 匹配文字和数字字符，也就是[A-Za-z0-9]，如：'G\w\*p'匹配以G后跟零个或多个文字或数字字符，然后是p。 |
| \W | \w的反置形式，匹配一个或多个非单词字符，如点号句号等。 |
| \b | 单词锁定符，如:'\bgrep\b'只匹配grep。 |

# 按照正则表达式查找

grep match\_pattern file 或 grep “match\_pattern” file

或

grep match\_pattern file1 file2 file3 … 或 grep “match\_pattern” file1 file2 file3 …

使用正则表达式-E选项

grep -E "[1-9]+"

# 或

egrep "[1-9]+"

# 输出除所有匹配行之外的所有行-v选项

grep –v “match\_pattern” file

# 只输出文件中匹配到的部分 -o 选项

grep –o “match\_pattern” file

# 统计文件或者文本中包含匹配字符串的行数 -c 选项：

grep -c "text" file\_name

# 递归(-r)查找当前目录下包含config字符串的所有文件名(-l)

grep –rl “config” ./

# 匹配config字符串时忽略大小写-i

grep –ril “config” ./

# 选项-e进行多个匹配样式

echo this is a text line | grep -e "is" -e "line" –o

is

line

#也可以使用 \*\*-f\*\* 选项来指定匹配样式文件，在样式文件中逐行写出需要匹配的字符。

cat patfile

aaa

bbb

echo aaa bbb ccc ddd eee | grep -f patfile –o

# 在grep查找的结果中包含或者排除指定的文件

# 只在目录中所有的.php和.html文件中递归搜索字符"main()"

grep "main()" . -r --include \*.{php,html}

# 在搜索结果中排除所有README文件

grep "main()" . -r --exclude "README"

# 在搜索结果中排除filelist文件列表里的文件

grep "main()" . -r --exclude-from filelist