

# Linux命令大全

当前位置： 首页 » 文件和目录管理 » grep

## grep命令

**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。

### 选项

- a 不要忽略二进制数据。
- A<显示列数> 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 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。
- c 计算符合范本样式的列数。
- C<显示列数>或-C<显示列数> 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> 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> 指定字符串作为查找文件内容的范本样式。
- E 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> 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 将范本样式视为固定字符串的列表。
- G 将范本样式视为普通的表示法来使用。
- h 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 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 忽略字符大小写的差别。
- l 列出文件内容符合指定的范本样式的文件名称。
- L 列出文件内容不符合指定的范本样式的文件名称。
- n 在显示符合范本样式的那一列之前，标示出该列的编号。
- q 不显示任何信息。
- R/-r 此参数的效果和指定“-d recurse”参数相同。
- s 不显示错误信息。
- v 反转查找。
- w 只显示全字符合的列。
- x 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。

### grep命令常见用法

在文件中搜索一个单词，命令会返回一个包含“match\_pattern”的文本行：

```
grep match_pattern file_name
grep "match_pattern" file_name
```

在多个文件中查找:

```
grep "match_pattern" file_1 file_2 file_3 ...
```

输出除之外的所有行 **-v** 选项:

```
grep -v "match_pattern" file_name
```

标记匹配颜色 **--color=auto** 选项:

```
grep "match_pattern" file_name --color=auto
```

使用正则表达式 **-E** 选项:

```
grep -E "[1-9]+"  
或  
egrep "[1-9]+"
```

只输出文件中匹配到的部分 **-o** 选项:

```
echo this is a test line. | grep -o -E "[a-z]+\."  
line.  
  
echo this is a test line. | egrep -o "[a-z]+\."  
line.
```

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
```

输出包含匹配字符串的行数 **-n** 选项:

```
grep "text" -n file_name  
或  
cat file_name | grep "text" -n  
  
#多个文件  
grep "text" -n file_1 file_2
```

打印样式匹配所位于的字符或字节偏移:

```
echo gun is not unix | grep -b -o "not"  
7:not
```

#一行中字符串的字符便宜是从该行的第一个字符开始计算, 起始值为0。选项 **-b -o** 一般总是配合使用。

搜索多个文件并查找匹配文本在哪些文件中:

```
grep -l "text" file1 file2 file3...
```

## grep递归搜索文件

在多级目录中对文本进行递归搜索:

```
grep "text" . -r -n  
# .表示当前目录。
```

忽略匹配样式中的字符大小写:

```
echo "hello world" | grep -i "HELLO"  
hello
```

选项 **-e** 制动多个匹配样式:

```
echo this is a text line | grep -e "is" -e "line" -o  
is  
line  
  
#也可以使用-f选项来匹配多个样式, 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搜索结果中包括或者排除指定文件:

```
#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
```

使用0值字节后缀的grep与xargs:

```
#测试文件:
echo "aaa" > file1
echo "bbb" > file2
echo "aaa" > file3

grep "aaa" file* -lZ | xargs -0 rm

#执行后会删除file1和file3, grep输出用-z选项来指定以0值字节作为终结符文件名 (\0), xargs -0 读取输入并用0
值字节终结符分隔文件名, 然后删除匹配文件, -z通常和-l结合使用。
```

grep静默输出:

```
grep -q "test" filename

#不会输出任何信息, 如果命令运行成功返回0, 失败则返回非0值。一般用于条件测试。
```

打印出匹配文本之前或者之后的行:

```
#显示匹配某个结果之后的3行, 使用 -A 选项:
seq 10 | grep "5" -A 3
5
6
7
8

#显示匹配某个结果之前的3行, 使用 -B 选项:
seq 10 | grep "5" -B 3
2
3
4
5

#显示匹配某个结果的前三行和后三行, 使用 -C 选项:
seq 10 | grep "5" -C 3
2
3
4
5
6
7
```

8

#如果匹配结果有多个, 会用"--"作为各匹配结果之间的分隔符:

```
echo -e "a\nb\nc\na\nb\nc" | grep a -A 1
```

a

b

--

a

b

---

## 相关命令

---

comm    fgrep    rev    look    fmt    uniq    tac    tee    col    colrm    wc    tr

split    spell    expand    join    diffstat    unexpand    ispell    paste

命令直达(输入完整命令)

[进入](#)

模糊搜索(输入关键词)

[搜索](#)

---

## Linux下载

---

[Ubuntu下载](#)

[CentOS下载](#)

[返回顶部](#) ↑

---

[Linux命令大全](#)

[关于/联系](#)

[收藏本站请使用Ctrl+D](#)

[Shell脚本攻略](#)

[Shell正则表达式](#)

[网站地图](#)

[共收录](#)

**597** 条Linux系统命令

在Linux命令大全 (man.linuxde.net) 可以查询您所需要的Linux命令教程和相关实例。如果您觉得本站内容对您有所帮助, 请推荐给更多需要帮助的人。