

Linux grep

1..主要参数

[options]主要参数:

- c: 只输出匹配行的计数。
- I: 不区分大小写(只适用于单字符)。
- h: 查询多文件时不显示文件名。
- l: 查询多文件时只输出包含匹配字符的文件名。
- n: 显示匹配行及行号。
- s: 不显示不存在或无匹配文本的错误信息。
- v: 显示不包含匹配文本的所有行。

pattern 正则表达式主要参数:

- \: 忽略正则表达式中特殊字符的原有含义。
- ^: 匹配正则表达式的开始行。
- \$: 匹配正则表达式的结束行。
- \<: 从匹配正则表达式的行开始。
- \>: 到匹配正则表达式的行结束。
- []: 单个字符, 如[A]即 A 符合要求。
- [-]: 范围, 如[A-Z], 即 A、B、C 一直到 Z 都符合要求。
- .: 所有的单个字符。
- *: 有字符, 长度可以为 0。

4.grep 命令使用简单实例

```
$ grep 'test' d*
```

显示所有以 d 开头的文件中包含 test 的行。

```
$ grep 'test' aa bb cc
```

显示在 aa, bb, cc 文件中匹配 test 的行。

```
$ grep '[a-z]\{5\}' aa
```

显示所有包含每个字符串至少有 5 个连续小写字母的字符串的行。

```
$ grep 'w(es)t.*\1' aa
```

如果 west 被匹配, 则 es 就被存储到内存中, 并标记为 1, 然后搜索任意个字符(.*) , 这些字符后面紧跟着另外一个 es(\1), 找到就显示该行。如果用 egrep 或 grep -E, 就不用” \”号进行转义, 直接写成’ w(es)t.*\1’就可以了。

5.grep 命令使用复杂实例

假设您正在’ /usr/src/[linux](#) /Doc’目录下搜索带字符串’ magic’的文件:

```
$ grep magic /usr/src/linux/Doc/*
```

```
sysrq.txt:* How do I enable the magic SysRQ key?
```

sysrq.txt:* How do I use the magic SysRQ key?

其中文件 'sysrq.txt' 包含该字符串，讨论的是 SysRQ 的功能。

默认情况下，'grep' 只搜索当前目录。如果此目录下有许多子目录，'grep' 会以如下形式列出：

```
grep: sound: Is a directory
```

这可能会使 'grep' 的输出难于阅读。这里有两种解决的办法：

明确要求搜索子目录： `grep -r`

或忽略子目录： `grep -d skip`

如果有很多输出时，您可以通过管道将其转到 'less' 上阅读：

```
$ grep magic /usr/src/linux/Documentation/* | less
```

这样，您就可以更方便地阅读。

有一点要注意，您必需提供一个文件过滤方式(搜索全部文件的话用 *)。如果您忘了，'grep' 会一直等着，直到该程序被中断。如果您遇到了这样的情况，按 <CTRL c>，然后再试。

下面还有一些有意思的命令行参数：

`grep -i pattern files`：不区分大小写地搜索。默认情况区分大小写，

`grep -l pattern files`：只列出匹配的文件名，

`grep -L pattern files`：列出不匹配的文件名，

`grep -w pattern files`：只匹配整个单词，而不是字符串的一部分(如匹配 'magic'，而不是 'magical')，

`grep -C number pattern files`：匹配的上下文分别显示[number]行，

`grep pattern1 | pattern2 files`：显示匹配 pattern1 或 pattern2 的行，

`grep pattern1 files | grep pattern2`：显示既匹配 pattern1 又匹配 pattern2 的行。

这里还有些用于搜索的特殊符号：

\< 和 \> 分别标注单词的开始与结尾。

例如：

`grep man *` 会匹配 'Batman'、'manic'、'man' 等，

`grep '\<man'*` 匹配 'manic' 和 'man'，但不是 'Batman'，

`grep '\<man\>'` 只匹配 'man'，而不是 'Batman' 或 'manic' 等其他的字符串。