**Grep学习笔记**

整理：Jims of [肥肥世家](http://www.ringkee.com/)

<[yjnet@21cn.com](mailto:yjnet@21cn.com)>

Copyright © 2004 本文遵从GPL协议，欢迎转载、修改、散布。

第一次发布时间：2004年7月16日

**Table of Contents**

1. grep简介

2. grep正则表达式元字符集（基本集）

3. 用于egrep和 grep -E的元字符扩展集

4. POSIX字符类

5. Grep命令选项

6. 实例

# 1. grep简介

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。Unix的grep家族包 括grep、egrep和fgrep。egrep和fgrep的命令只跟grep有很小不同。egrep是grep的扩展，支持更多的re元字符， fgrep就是fixed grep或fast grep，它们把所有的字母都看作单词，也就是说，正则表达式中的元字符表示回其自身的字面意义，不再特殊。linux使用GNU版本的grep。它功能 更强，可以通过-G、-E、-F命令行选项来使用egrep和fgrep的功能。

grep的工作方式是这样的，它在一个或多个文件中搜索字符串模板。如果模板包括空格，则必须被引用，模板后的所有字符串被看作文件名。搜索的结果被送到屏幕，不影响原文件内容。

grep可用于shell脚本，因为grep通过返回一个状态值来说明搜索的状态，如果模板搜索成功，则返回0，如果搜索不成功，则返回1，如果搜索的文件不存在，则返回2。我们利用这些返回值就可进行一些自动化的文本处理工作。

# 2. grep正则表达式元字符集（基本集）

^

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。

# 3. 用于egrep和 grep -E的元字符扩展集

+

匹配一个或多个先前的字符。如：'[a-z]+able'，匹配一个或多个小写字母后跟able的串，如loveable,enable,disable等。

?

匹配零个或多个先前的字符。如：'gr?p'匹配gr后跟一个或没有字符，然后是p的行。

a|b|c

匹配a或b或c。如：grep|sed匹配grep或sed

()

分组符号，如：love(able|rs)ov+匹配loveable或lovers，匹配一个或多个ov。

x{m},x{m,},x{m,n}

作用同x\{m\},x\{m,\},x\{m,n\}

# 4. POSIX字符类

为 了在不同国家的字符编码中保持一至，POSIX(The Portable Operating System Interface)增加了特殊的字符类，如[:alnum:]是A-Za-z0-9的另一个写法。要把它们放到[]号内才能成为正则表达式，如[A- Za-z0-9]或[[:alnum:]]。在linux下的grep除fgrep外，都支持POSIX的字符类。

[:alnum:]

文字数字字符

[:alpha:]

文字字符

[:digit:]

数字字符

[:graph:]

非空字符（非空格、控制字符）

[:lower:]

小写字符

[:cntrl:]

控制字符

[:print:]

非空字符（包括空格）

[:punct:]

标点符号

[:space:]

所有空白字符（新行，空格，制表符）

[:upper:]

大写字符

[:xdigit:]

十六进制数字（0-9，a-f，A-F）

# 5. Grep命令选项

-?

同时显示匹配行上下的？行，如：grep -2 pattern filename同时显示匹配行的上下2行。

-b，--byte-offset

打印匹配行前面打印该行所在的块号码。

-c,--count

只打印匹配的行数，不显示匹配的内容。

-f File，--file=File

从文件中提取模板。空文件中包含0个模板，所以什么都不匹配。

-h，--no-filename

当搜索多个文件时，不显示匹配文件名前缀。

-i，--ignore-case

忽略大小写差别。

-q，--quiet

取消显示，只返回退出状态。0则表示找到了匹配的行。

-l，--files-with-matches

打印匹配模板的文件清单。

-L，--files-without-match

打印不匹配模板的文件清单。

-n，--line-number

在匹配的行前面打印行号。

-s，--silent

不显示关于不存在或者无法读取文件的错误信息。

-v，--revert-match

反检索，只显示不匹配的行。

-w，--word-regexp

如果被\<和\>引用，就把表达式做为一个单词搜索。

-V，--version

显示软件版本信息。

# 6. 实例

要用好grep这个工具，其实就是要写好正则表达式，所以这里不对grep的所有功能进行实例讲解，只列几个例子，讲解一个正则表达式的写法。

$ ls -l | grep '^a'

通过管道过滤ls -l输出的内容，只显示以a开头的行。

$ 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个连续小写字符的字符串的行。

$ 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。