正则表达式基础

内容提要

1. 掌握正则表达式的使用
2. 学会使用 grep 过滤文本

正则表达式基础

何谓正则表达式

正则表达式是使用某种模式（pattern）去匹配（matching）一类字符串的一个公式。通常使用正则表达式进行查找、替换等操作。虽然复杂的正则表达式对于初学者来说晦涩难懂，但对于 Linux 使用者来说，学会正则表达式的使用是非常必要的。在适当的情况下使用正则表达式可以极大地提高工作效率。

有两种风格的正则表达式：

- POSIX 风格的正则表达式
- Perl 风格的正则表达式（Perl-compatible regular expression）

支持 RE 的文本处理工具

本节主要介绍 POSIX 风格的正则表达式及其工具的使用。下面列出 Linux 下常用的支持 POSIX 风格正则表达式的工具：

1. 基本的正则表达式 Basic regular expression (BRE)
   - grep 按模式匹配文本
   - ed 一个原始的行编辑器
   - sed 一个流编辑器
   - vim 一个屏幕编辑器
   - emacs 一个屏幕编辑器
2. 扩展的正则表达式 Extended regular expression (ERE)
   - egrep 按模式匹配文本
   - awk 进行简单的文本处理

正则表达式的组成

正则表达式由一些普通字符和一些元字符（metacharacters）组成。普通字符包括大小写的字母、数字（即所有非元字符），而元字符则具有特殊的含义。

下面列出 POSIX RE 的元字符及其含义：

1、POSIX RE 用于方括号之外的元字符

特殊字符	含义	类型	举例	说明
^	匹配首字符	BRE	^x	以字符x开始的字符串

$	匹配尾字符	BRE	x$	以x字符结尾的字符串
.	匹配任意一个字符	BRE	l..e	love, life, live …
?	匹配任意一个可选字符	**ERE**	xy?	x, xy
*	匹配零次或多次重复	BRE	xy*	x, xy, xyy, xyyy …
+	匹配一次或多次重复	**ERE**	xy+	xy, xyy, xyyy …
[…]	匹配任意一个字符	BRE	[xyz]	x, y, z
()	对正则表达式分组	**ERE**	(xy)+	xy, xyxy, xyxyxy, …
\{n\}	匹配n次	BRE	go\{2\}gle	google
\{n,\}	匹配最少n次	BRE	go\{2,\}gle	google, gooogle, goooogle …
\{n,m\}	匹配n到m次	BRE	go\{2,4\}gle	google, gooogle, goooogle
{n}	匹配n次	**ERE**	go{2}gle	google
{n,}	匹配最少n次	**ERE**	go{2,}gle	google, gooogle, goooogle …
{n,m}	匹配n到m次	**ERE**	go{2,4}gle	google, gooogle, goooogle
|	以或逻辑连接多个匹配	**ERE**	good|bon	匹配 good 或 bon
\	转义字符	BRE	*	*

2、POSIX RE 用于方括号之内的元字符

特殊字符	含义	类型	举例	说明
^	非（仅用于启始字符）	BRE	[^xyz]	匹配xyz之外的任意一个字符
-	用于指明字符范围（不能是首字符和尾字符）	BRE	[a-zA-Z]	匹配任意一个字母
\	转义字符	BRE	[\.]	.

grep

grep 简介

grep [http://www.proxyserve.net/index.php?q=aHR0cDovL21hbi5jeC9ncmVw] (global search regular expression) 是一个强大的文本搜索工具。grep 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：

- grep 使用 Basic regular expression (BRE) 书写匹配模式
- egrep 使用 Extended regular expression (ERE) 书写匹配模式，等效于 grep -E
- fgrep 不使用任何正则表达式书写匹配模式（以固定字符串对待），执行快速搜索，等效于 grep -F

grep 命令的格式

grep 命令的格式如下：

```
grep [options] PATTERN [FILE...]
```

其中：

- PATTERN 是用正则表达式书写的模式
- FILE 是要查找的文件，可以是用空格间隔的多个文件，省略时表示在标准输入查找
- 常用的参数：
  - **-c**：只显示匹配行的次数
  - **-i**：搜索时不区分大小写

- **-n**：输出匹配行的行号
- **-v**：输出不匹配的行（反向选择）
- **-A NUM**：同时输出匹配行的后 NUM 行
- **-B NUM**：同时输出匹配行的前 NUM 行
- **-C NUM**：同时输出匹配行的前、后各 NUM 行

- FILE 可是使用 Shell 的通配符在多个文件中查找 PATTERN
- 通常必须用单引号将整个模式 PATTERN 括起来
- grep 命令不会对输入文件进行任何修改或影响，可以使用输出重定向将结果存为文件

grep 使用举例

```
# 在文件 myfile 中查找包含指定的字符串的行
# 如果使用 grep 命令查找指定的字符串（不使用正则表达式），PATTERN 可以不用单引号括起来
$ grep osmond myfile
$ fgrep osmond myfile
# 在文件 myfile1 myfile2 myfile3 中查找包含指定的字符串的行
$ grep 'osmond' myfile1 myfile2 myfile3
$ fgrep 'osmond' myfile1 myfile2 myfile3
# 在当前目录的所有文件中查找包含指定的字符串的行
$ grep osmond *
$ fgrep osmond *
# 在当前目录的所有文件中查找包含字符 * 的行
$ grep '*' *
$ fgrep '*' *
# 在文件 myfile 中查找包含字符 $ 的行
# 为了强制 shell 将 \$（单反斜杠和美元符号）传递给 grep 命令，必须要使用 \\（双反斜杠）。
# \（单反斜杠）字符可通知 grep 命令将其后的字符（本例中为 $）视作原义字符而不是元字符。
# 如果使用 fgrep 命令，则可以不必使用反斜杠之类的转义字符。
$ grep \\$  myfile
$ grep '\$'  myfile
$ fgrep '$'  myfile
$ fgrep $  myfile
# 匹配 myfile 中所有的空行
$ grep  '^$'  myfile
# 显示 myfile 中第一个字符为字母的所有行
$ grep  '^[a-zA-Z]'  myfile
# 在文件 myfile 中查找首字符不是 # 的行（即过滤掉注释行）
$ grep -v '^#' myfile
# 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 '[a-z]\{5\}' myfile
$ egrep '[a-z]{5}' myfile
# 在文件 myfile 中查找包含日期格式（形如：yyyy-mm-dd）的行
$ grep '[12][0-9]\{3\}-[01][0-9]-[0-3][0-9]' myfile
$ egrep '[12][0-9]{3}-[01][0-9]-[0-3][0-9]' myfile
# 在文件 myfile 中查找与abc 或 xyz 字符串匹配的所有行
$ egrep 'abc|xyz' myfile
# 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。
$ grep 'w\(es\)t.*\1' myfile
$ egrep 'w(es)t.*\1' myfile
# 通过管道过滤ls输出的内容，只显示以 l 开头的行
$ ls -l| grep '^l'
# 通过管道过滤ls输出的内容，只显示以可写的文件或目录的行
$ ls -l| grep '[-d].w..w..w.'
# 通过管道过滤ls输出的内容，只显示以 ~ 或 - 或 .bak 结尾的行
$ ls | egrep '(~|-|\.bak)$'
```

参考

- http://net.pku.edu.cn/~yhf/tao_regexps_zh.html [http://www.proxyserve.net/index.php?q=aHR0cDovL25ldC5wa3UuZWR1LmNuL355aGYvdGFvX3JlZ2V4cHNfemguaHRtbA%3D%3D]

- http://www.ringkee.com/note/opensource/grep.htm [http://www.proxyserve.net/index.php?q=aHR0cDovL3d3dy5yaW5na2VlLmNvbS9ub3RlL29wZW5zb3VyY2UvZ3JlcC5odG0%3D]
- http://www.regexlab.com/zh/ [http://www.proxyserve.net/index.php?q=aHR0cDovL3d3dy5yZWdleGxhYi5jb20vemgv]
- 显示源文件
- 登录