正则表达式

1. 什么是正则表达式？

   正则表达式是一种符号表示法。用于识别文本模式。许多命令行工具和大多数编程语言都支持正则表达式，以此来解决文本操作方面的问题。然而在不同的工具、语言之间的正则表达式都略有不同。我们在此处讨论的是 POSIX 兼容的正则表达式。

2. grep——文本搜索

   `grep [options] regex [file…]`

   其中的 regex 表示正则表达式

options	功能描述
-I	忽略大小写
-v	输出不匹配行。正常情况下 grep 是输出匹配行
-c	输出匹配项的数目，而不是输出匹配项自身
-l	输出匹配项文件名而不是输出匹配行自身
-L	与-l 类似，输出的是不包含匹配项的文件名
-n	在每个匹配行的前面加上行号
-h	进行多文件搜索时，抑制文件名输出

3. 元字符和文字

   grep 中进行搜索时实际上一直都在使用正则表达式。只是之前采用的都是简单的文本。除了普通文本外，正则表达式还支持更为复杂的元字符，这些元字符包括：

^	$	.	[	]	{	}	-	?	*	+	(	)	|	\

   这些元字符出现在 shell 中时都具有特殊的含义，所以在包含它们时，应该用单引号包含起来。

4. 任意字符——圆点

   圆点.用于匹配任意一个字符。

5. 锚

   ^和$是锚。正则表达式只与行的开头^或末尾$的内容进行匹配比较。

   `grep -h '^zip' dirlist*.txt` 查找 zip 开头的行

   `grep -h 'zip$' dirlist*.txt` 查找 zip 结尾的行

   `grep -h '^zip$' dirlist*.txt` 查找行内只包含 zip 的内容

   正则表达式^$用于匹配一个空行

6. 中括号表达式和字符类

   中括号除了可以用于匹配正则表达式中给定位置的任意字符外，还可以用于匹配指定字符集中的单个字符。

   `grep -h '[bg]zip' dirlist*.txt` 该命令查找所有包含 bzip 和 gzip 的文本行

   如果在[]的起始处插入一个^，那么则表示否定。即剩下的字符不应该出现，如[^bg]zip

   -表示范围，如[a-z0-9A-Z]表示任意一个大小写字母和数字。

   如果-本身也要作为可以被选择内容，则应该使用-开头，如[-a-z0-9A-Z]

7. 传统字符范围和 POSIX 字符类

早期的 UNIX 系统使用了 ASCII 字符集。ASCII 使用了类似如下排序 ABC..XYZabc...xyz。之后的 UNIX 标准 POSIX 中规定使用字典顺序进行排序，即 aAbBcC...xXyYzZ。所以当存在如下内容即[A-Z]时，传统 ASCII 使用 ASCII 顺序表示全大写字母，POSIX 得到的是除了字母 a 之外的字母。

可以通过 echo $LANG 查询系统中采用的语言。$LANG 环境变量包含语言的名称以及字符集。

locale 命令可以查看系统中与语言相关的各环境变量的设置

如果想要改成 POSIX 标准方式，可以采用 export LANG=POSIX

POSIX 字符集

字符类	描述
[:alnum:]	字母字符和数字字符，在 ASCII 中，使用[A-Za-z0-9]
[:word:]	与[:alnum:]一样，只是多了下划线字符
[:alpha:]	表示大小写字母
[:blank:]	表示空白，表示空格和制表符(\t)
[:cntrl:]	ASCII 中的控制码，即不打印字符，如 Esc 等，即 ASCII 中的 0-31 和 127
[:digit:]	表示 0-9
[:graph:]	表示可见字符，即 ASCII 字符集中的 33-126 号字符
[:lower:]	表示小写字母
[:upper:]	表示大写字母
[:punct:]	表示标点符号
[:print:]	可打印字符，等价于[:graph:]加上空格
[:space:]	表示空格，制表符，回车符，换行符，垂直制表符以及换页符[\t\r\n\v\f]
[:xdigit:]	用于表示十六进制的字符，即 ASCII 中的[0-9A-Fa-f]

例子： `ls /usr/sbin/[[:upper:]]*`

8. 扩展正则表达式

POSIX 规范中将正则表达式的实现方法分为 BRE（基本正则表达式）和 ERE（扩展正则表达式）。

在 ERE 中又加入了( ) { } ? + |等元字符

- 或选项 |

  ```
  echo "AAA" | grep -E 'AAA|BBB|CCC'
  ```

  -E 表示让 grep 命令支持 ERE。grep -E 等价于 egrep 命令
- 限定符 ?

  ?用于匹配某元素 0 次或者 1 次。

  ```
  echo "(555) 123-4567" | grep -E '^\(?[0-9][0-9][0-9]\)?'
  ```
- *

  *用于匹配某元素 0 次或多次
- +

  用于匹配某元素 1 次或多次
- {}

  {} 用于指定匹配某元素的次数

指定项	含义
{n}	前面的元素出现 n 次
{n， m}	前面的元素出现 n-m 次
{n，}	前面的元素至少出现 n 次
{， m}	前面的元素最多出现 m 次

例子：`echo "(555) 123-4567" | grep -E '^\(?[0-9]{0,3}\)? [0-9]{3}-[0-9]{4}$'`