

🔗 正则表达式基础 (/courses/90)

# grep 命令与正则表达式

## 一、实验介绍

### 1.1 实验内容

本次实验将介绍通过 grep 命令使用正则表达式。

### 1.2 实验知识点

- grep 命令
- 字符组匹配
- 行首行尾符
- 任意和重复字符
- 限定连续字符范围

### 1.3 实验环境

课程使用的实验环境为 Ubuntu Linux 14.04 64 位版本。实验中会用到程序：

- Xfce终端

## 二、使用步骤

接下来开始具体实验。

### 2.1 搜寻特定字符串"the"

参数说明：

- a：将 binary 档案以 text 档案的方式搜寻数据
- c：计算找到 '搜寻字符串' 的次数
- i：忽略大小写的不同，所以大小写视为相同
- n：顺便输出行号
- v：反向选择，亦即显示出没有 '搜寻字符串' 内容的那一行！

正则表达式基础 (regular\_express780).txt

```
$ grep -in 'the' regular_express.txt
```

操作截图：

```
shiyanolou:~/ $ grep -n 'the' regular_express.txt
8:I can't finish the test.
12:the symbol '*' is represented as start.
15:You are the best is mean you are the no. 1.
16:The world <Happy> is the same with "glad".
18:google is the best tools for search keyword.
```

```
shiyanolou:~/ $ grep -vn 'the' regular_express.txt
9:Oh! The soup taste good.
14:The gd software is a library for drafting programs.
15:You are the best is mean you are the no. 1.
16:The world <Happy> is the same with "glad".
18:google is the best tools for search keyword.
```

反向查找,也就是说当该行没有" the " 字符时才显示在屏幕上。

```
$ grep -vn 'the' regular_express.txt
```

## 2.2 字符组匹配:

[]中包含的任意一个字符。只能是一个。

字符组支持由连字符 " - " 来表示一个范围。当 " - " 前后构成范围时, 要求前面字符的码位小于后面字符的码位。

[^...] 排除型字符组。排除后面的字符。

[abc]	: 表示"a"或"b"或"c"
[0-9]	: 表示 0~9 中任意一个数字, 等价于[0123456789]
[\u4e00-\u9fa5]	: 表示任意一个汉字
[^a1<]	: 表示除"a"、"1"、"<"外的其它任意一个字符
[^a-z]	: 表示除小写字母外的任意一个字符

查找"tast"或者"test"两个字符串。

```
# grep -n 't[ae]st' regular_express.txt
```

查找不包含"#"的字符串

```
# grep -n '^[^#]' regular_express.txt
```

操作截图：

```
shiyanolou:~/ $ grep -n 't[ae]st' regular_express.txt [17:31:11]
8:正则表达式基础 (courses/90)
9:Oh! The soup taste good.
shiyanolou:~/ $ [17:31:14]
```



查找字符“oog”，如果我不想要“oog”字符前面有g，则查找“`^[^g]oog`”，同理，查找“`^[^go]oog`”是指字符“oog”前面不能为g或者o。

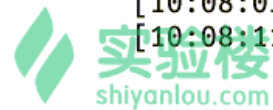
```
$ grep -n 'oog' regular_express.txt

$ grep -n '^[^g]oog' regular_express.txt

$ grep -n '^[^go]oog' regular_express.txt
```

操作截图：

```
shiyanolou:~/ $ grep -n 'oog' regular_express.txt [10:07:56]
18:google is the best tools for search keyword.
19:gooooooogle yes!
shiyanolou:~/ $ grep -n '^[^g]oog' regular_express.txt [10:08:01]
19:gooooooogle yes!
shiyanolou:~/ $ grep -n '^[^go]oog' regular_express.txt [10:08:03]
shiyanolou:~/ $ [10:08:11]
```



`grep -n '^[^go]oog' regular_express.txt` 结果返回为空，表明没有匹配到满足要求的字符串。

## 2.3 行首符: ^ 与行尾符: \$

在第一个小实验中，查找了一行字符中含有“the”的，如果你想要只查找行首为“the”的字符行，则使用以下命令：

```
$ grep -n '^the' regular_express.txt
```

查找行首为大写字母的所有行：

'`^[A-Z]`' 表示以大写字母开头。

'`^[^A-Z]`' 表示除了大写字母 A-Z 的所有字符。

```
# grep -n '^[A-Z]' regular_express.txt
```

查找以 d 字母结尾的行：

正则表达式基础(regular\_express.txt)

操作截图：

```
shiyancelou:~/ $ grep -n 'd$' regular_express.txt [10:32:30]
21:# I am VBird
shiyancelou:~/ $ grep -n '^the' regular_express.txt [10:32:33]
12:the symbol '*' is represented as start.
shiyancelou:~/ $ [10:32:39]
```

I



查找空行：

```
$ grep -n '^$' regular_express.txt
```

应用实例：

查看/etc/insserv.conf 文档

'^\$': 过滤掉空白行

'^#': 过滤掉注释行（以#号开头）

```
$ cat -n /etc/insserv.conf

$ grep -v '^$' /etc/insserv.conf | grep -v '^#'
```

## 2.4 任意一个字符: "." (小数点) 与重复字符 "\*" (星号)

查找 a?ou? 类型的字符。

```
$ grep -n 'a.ou.' regular_express.txt
```

'a.ou.': 小数点表示任意一个字符，一个小数点只能表示一个未知字符。

\* (星号): 代表重复前面 0 个或者多个字符。  
e\*: 表示具有空字符或者一个以上 e 字符。  
ee\*, 表示前面的第一个 e 字符必须存在。第二个 e 则可以是 0 个或者多个 e 字符。  
eee\*, 表示前面两个 e 字符必须存在。第三个 e 则可以是 0 个或者多个 e 字符。  
ee\*e: 表示前面的第一个与第三个 e 字符必须存在。第二个 e 则可以是 0 个或者多个 e 字符。

下面的第一条命令与第二条命令由于允许存在空字符，所以会打印所有文本。

## 正则表达式基础(courses790).txt

```
$ grep -n '@*' regular_express.txt
```

```
$ grep -n 'eee*' regular_express.txt
```

## 2.5 限定连续字符范围{ }

{ }可限制一个范围区间内的重复字符数。举个例子，若要找出 2~5 个 o 的连续字符串，如何做？此时便要用到{ }了。由于 { 与 } 在 shell 中有特殊意义，需要用到转义字符\。

查找连续的两个 o 字符：

```
$ grep -n 'o\{2\}' regular_express.txt
```

结果与命令 `grep -n 'ooo*' regular_express.txt` 的结果相同。

查找 g 后面接 2 到 5 个 o，然后再接 g 的字符串

```
$ grep -n 'go\{2,5\}g' regular_express.txt
```

总结：

`^word` 表示带搜寻的字符串(word)在行首

`word$` 表示带搜寻的字符串(word)在行尾

`.`(小数点) 表示 1 个任意字符

`\` 表示转义字符，在特殊字符前加\会将特殊字符意义去除

`*` 表示重复 0 到无穷多个前一个 RE(正则表达式)字符

`[list]` 表示搜索含有 l,i,s,t 任意字符的字符串

`[n1-n2]` 表示搜索指定的字符串范围,例如[0-9] [a-z] [A-Z]等

`[^list]` 表示反向字符串的范围,例如[^0-9]表示非数字字符, [^A-Z]表示非大写字母范围

`\{n,m\}` 表示找出 n 到 m 个前一个 RE 字符

`\{n,\}` 表示 n 个以上的前一个 RE 字符

## 三、实验总结

本节实验介绍了如何使用 grep 命令与正则表达式对文本字符串进行搜索。下一节我们将进入 sed 工具命令的学习。

## 四、参考链接

正则表达式基础 (/courses/90)

---

《鸟哥的 linux 私房菜》 鸟哥著 王世江 改编

---

*\*本课程内容，由作者授权实验楼发布，未经允许，禁止转载、下载及非法传播。*

上一节：基础正则表达式介绍与练习 (/courses/90/labs/779/document)

下一节：正则表达式运用之 sed工具命令 (/courses/90/labs/781/document)