

## Shell 编程

---

### 一、正则表达式

### 二、Shell 编程中常用命令

### 三、shell 条件测试

### 四、熟悉 shell 中的常用语法（流程控制）

(1)if (2)while (3)for (4)case

### 五、熟悉 shell 中函数的使用

---

### 一、正则表达式

...

1 ^

只匹配行首

2 \$

只匹配行尾

3 \*

匹配 0 个或者多个单字符

4 []

只匹配[]内字符，可以是一个单字符，也可以是字符序列，可以使用\*表示[]内字符序列范围，如用[1-5]代替[12345]

5 \

只用来屏蔽一个元字符的特殊含义

6 .

只匹配任意单字符

7 pattern\{n\}

匹配 n 次 pattern

8 pattern\{n,\}

匹配 n 次以上 pattern

9 pattern\{n,m\}

匹配 n 到 m 次 pattern

11 ^只允许在一行的开始匹配字符或单词

^d 筛选出以 d 开头的文件属性

12 ^\$

匹配空行

13 ^.\$

匹配包含一个字符的行

14 kkk\$

匹配以 kkk 结尾的所有字符

15 \\*\.pas

匹配以 \*.pas 结尾的所有字符或文件

16 a\{2\}b

a 出现两次, aab

17 a\{4,\}b

a 至少出现 4 次, aaaab, aaaaab ..

18 a\{2,4\}

a 出现次数范围 2-4 次

19 [0-9]\{3\}\.[0-9]\{3\}\.[0-9]\{3\}\.[0-9]\{3\}

匹配所有 ip 地址

```

## 二、Shell 编程中常用命令

```

行提取命令

grep 选项:            -v        -n        -i

grep "[^a-z]oo" aa (文件名)

oo 前不是小写字母的行匹配。 注意: 和开头没有关系

grep "\.\$" aa

匹配以. 结尾的行

grep "^[^A-Za-z]" aa

匹配不以字母开头的行            注意: 所有字母不能这样写    A-z

grep "^\$" aa

匹配空白行

grep "oo\*" aa

匹配最少一个 o

grep "g.\*d" aa

匹配 g 开头, d 结尾, 中间任意字符    gd

列提取命令

awk '条件 {动作}'

last | awk '{printf \$1 "\t" \$3 "\n"}'

提取 last 显示结果的第一和第三列

```
last | grep "[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}" | awk '{printf $1  
"\t" $3 "\n"}
```

在 last 中提取包含 ip 的行，在行中提取第一和第三列

awk 内置变量 FS 指定分隔符

```
more /etc/passwd | awk 'BEGIN {FS=":"} {printf $1 "\t" $3 "\n"}
```

读取 passwd 文件，以 ":" 为分隔符，截取第一和第三列

BEGIN 在截取前使分隔符生效。如果没有 BEGIN，那么第一行自定义的分隔符不生效

```
1 last > file
```

把 last 命令结果保存在 file 文件中

```
2 awk '{print $0 "\n"}' file
```

查找出 file 文件中的每 1 列

```
3 awk '{print $1"\t"$7 "\n"}' file
```

查找出 file 文件中的第 1 列和第 7 列

```
```\n
```