

通配符

2019年5月31日 莫宇剑 符著 0:24

通配符：当在使用命令行时，有很多时间都用来查找你需要的文件，如ls find等。Shell提供了一套完整的字符串模式匹配规则，或者称之为元字符，当shell遇到上述字符时，就会把它们当作特殊字符，而不是文件名中的普通字符，这样用户就可以用它们来匹配相应的文件名。

通配符与正则表达式是有区别的，简单来说：通配符是用来通配的，正则表达式是用来匹配字符串的；在文本过滤工具里，都是用正则表达式，比如像awk，sed，等，是针对文件的内容的。而通配符多用在文件名上，比如查找find，ls，cp，等等。

Shell常见的通配符

```
| # 管道符
[root@test ~]# ls -al |wc
> # 输出重定向
>> # 输出追加重定向
< # 输入重定向
<< # 追加输入重定向
~ # 当前用户的家目录
`` # 引用命令被执行后的结果
[root@test ~]# echo date
date
[root@test ~]# echo `date`
2019年 06月 01日 星期六 16:52:55 CST
```

* 匹配全部字符，通配符

测试数据

```
[root@test ~]# touch a a6.log abc.log ac.txt b c c5.txt x.log A
```

例：查询以“.log”结尾的文件

```
[root@test ~]# ll *.log
```

```
[root@localhost test]# ll *.log
-rw-r--r--. 1 root root 0 Mar 21 16:49 a6.log
-rw-r--r--. 1 root root 0 Mar 21 16:38 abc.log
-rw-r--r--. 1 root root 0 Mar 21 16:38 x.log
```

<http://www.cnblogs.com/chenmh/>
技术交流群：62122135

find命令主要用于目标的搜索，尽量做到少使用，因为find会消耗大量的系统资源。使用该命令时，需要避开服务器运行高峰期，最好在指定的小范围内进行搜索，不要轻易使用全盘搜索。

```
[root@test ~]# find / -name init* -a -type f -exec ls -lh {} \;
```

```
[root@test ~]# find / -size -10k -exec ls -l {} \;
```

find参数详解：

-a # 根据多个条件进行匹配

-type # 根据文件类型进行匹配 “f”：表示文件、“d”：表示目录、“l”表示软件链接、“h”：表示硬链接

-exec # 对搜索结果进行操作

-exec # 命令 {} \;

```
[root@test ~]# find / -name init* -a -type f -exec ls -lh {} \;
```

```
[root@test ~]# find /etc -name *conf -a -type f -exec ls -lh {} \;
```

```
[root@test ~]# find /etc -name *log.conf -a -type f -exec ls -lh {} \;
```

? # 任意一个字符，通配符

```
[root@test ~]# ls /sbin/????
/sbin/arpd /sbin/genl /sbin/lshw /sbin/pppd /sbin/rngd /sbin/tcpd /sbin/vipw
/sbin/chat /sbin/halt /sbin/lsof /sbin/pvck /sbin/rtpd /sbin/tcsd
/sbin/crda /sbin/efup /sbin/mkfs /sbin/pwck /sbin/smbd /sbin/vgck
/sbin/fsck /sbin/init /sbin/nmbd /sbin/quot /sbin/sshd /sbin/vigr
[root@test ~]# ls /sbin/???
/sbin/arp /sbin/cbq /sbin/lid /sbin/lvm /sbin/mtr /sbin/sln /sbin/zic
/sbin/atd /sbin/gdm /sbin/lpc /sbin/lvs /sbin/pvs /sbin/vgs
[root@test ~]# ls /sbin/??
/sbin/ip /sbin/iw /sbin/sa /sbin/ss /sbin/tc
[root@test ~]#
```

[] # 代表“[”和“]”之间的某一个字符，比如[0-9]可以代表0-9之间的任意一个数字，[a-zA-Z]可以代表a-z和A-Z之间的任意一个字母，字母区分大小写。

例：只查询字母文件

```
[root@test ~]# ll [a-zA-Z]
```

```
[root@localhost test]# ll [a-zA-Z]
-rw-r--r--. 1 root root 0 Mar 21 16:38 a
-rw-r--r--. 1 root root 0 Mar 21 17:36 A
-rw-r--r--. 1 root root 0 Mar 21 16:38 b
-rw-r--r--. 1 root root 0 Mar 21 16:38 c
```

<http://www.cnblogs.com/chenmh/>
技术交流群：62122135

[abcd]表示匹配中括号内任意一个字符就成立

```
[root@test ~]# touch tast tbst tcst tsst tst
```

多通配符结果对比

```
[root@test ~]# ls t[abcd]st
```

```
tast tbst tcst
```

```
[root@test ~]# ls t*st
```

```
tast tbst tcst tsst tst
```

```
[root@test ~]# ls t?st
```

```
tast tbst tcst tsst
```

注释

& # 让程序或脚本切换到后台执行

&& # 并且，同时成立

```
[root@test ~]# date && time
2019年 06月 01日 星期六 17:10:04 CST
```

```
real 0m0.000s
user 0m0.000s
sys 0m0.000s
```

```
[root@test ~]# date && timea
2019年 06月 01日 星期六 17:10:06 CST
```

```
bash: timea: 未找到命令...
```

```
相似命令是: 'time'
```

```
[root@test ~]# datea && time
```

```
bash: datea: 未找到命令...
```

```
相似命令是: 'date'
```

|| # 在前一个命令结束时，若返回值为 false，继续执行下一个命令

;& # 在前一个命令结束时，而忽略其返回值，继续执行下一个命令

[^abcd] !、^ # 表示非，取反

[!abcd] !、^ # 表示非，取反

```
[root@test ~]# ls
anaconda-ks.cfg  chen02  chen05  chen08
chen00           chen03  chen06  chen09
chen01           chen04  chen07  initial-setup-ks.cfg
公共  图片  音乐
模板 文档 桌面
视频 下载
```

```
[root@test ~]# ls chen0[ ^7]
chen00  chen01  chen02  chen03  chen04  chen05  chen06  chen08  chen09
```

```
[root@test ~]# ls chen0[ !7]
chen00  chen01  _chen02  chen03  chen04  chen05  chen06  chen08  chen09
```

```
[root@test ~]# touch oo xx
[root@test ~]# find ./ ! -name oo
[root@test ~]# find ./ -name oo
```

{ } # 表示符合括号内包含的多个文件或产生一个序列

例：查询 ‘.log’ 和 “.txt” 结尾的文件

```
[root@test ~]# ll {*.log,*.txt}
```

```
[root@localhost test]# ll {*.log,*.txt}
-rw-r--r--. 1 root root 0 Mar 21 16:49 a6.log
-rw-r--r--. 1 root root 0 Mar 21 16:38 abc.log
-rw-r--r--. 1 root root 0 Mar 21 16:38 ac.txt
-rw-r--r--. 1 root root 0 Mar 21 16:38 a.txt
-rw-r--r--. 1 root root 0 Mar 21 16:38 c5.txt
-rw-r--r--. 1 root root 0 Mar 21 16:38 x.log
[root@localhost test]#
```

<http://www.cnblogs.com/chenmh/>
技术交流群：62122135

删除操作

例如：删除a、b、c和以.txt结尾的文件

```
[root@test ~]# rm -f {[abc],*.txt}
```

```
[root@localhost test]# rm -f {[abc],*.txt}
[root@localhost test]# ll
total 0
-rw-r--r--. 1 root root 0 Mar 21 17:36 A
-rw-r--r--. 1 root root 0 Mar 21 16:49 a6.log
-rw-r--r--. 1 root root 0 Mar 21 16:38 abc.log
-rw-r--r--. 1 root root 0 Mar 21 16:38 x.log
[root@localhost test]#
```

<http://www.cnblogs.com/chenmh/>
技术交流群：62122135

产生序列

```
[root@test ~]# touch {a..z}
```

```
[root@test ~]# ls {a..c}
```

```
[root@test ~]# ls [a-c]
```

```
a b c
```

```
[root@test ~]# ls [a-b]
```

```
a b
```

```
[root@test ~]# ls [a-z]
```

```
a b c d e f g h i j k l m n o p q r s t u v w x y z
```

```
[root@test ~]# touch test{A..Z}{A..Z}{a..z}
```

```
[root@test ~]# rm -rf test*
```

```
[root@test ~]# rm -rf {a..z}
```

```
[root@test ~]# rm -rf {A..Z}
```

```
[root@test ~]# touch {a,ae,ac}
```

利用 {} 来备份

```
[root@test ~]# cp a {e,fff} # 将ae复制一份叫afff
```

```
[root@test ~]# cp {a,fff} # 将a复制一份叫afff
```

```
[root@test ~]# rm -rf a?
```

```
[root@test ~]# cp a{,.bak} # 备份的简写
```

. # 当前目录

.. # 上级目录

```
[root@server ~]# find / -name hf
```

搜索根目录下的名称为hf的文件

```
[root@server ~]# find / -iname hf
```

搜索根目录下名称为hf的文件，忽略文件名大小写

```
[root@server ~]# find / -user tom -group root 搜索根目录下文件所有者是tom, 所属组是root
[root@server ~]# find / -size +900M -size -1000M 搜索根目录下的文件大小在900-1g之间的
[root@server ~]# find /tmp -cmin -15 搜索tmp目录下15分钟以内创建的文件
[root@server ~]# find / -perm -222 表示文件的权限必须u, g, o里面都有w的权限r, x可有可无
[root@server ~]# find / -perm +222 表示问的权限u, g, o里面有一个包含w的权限即可
[root@server ~]# find / -perm 222 查找权限为222的文件或目录
```

参考文献:

[1] linux通配符和正则表达式 <https://blog.csdn.net/youmatterhsp/article/details/80528761>