

Shell常用函数

- grep

grep在匹配通配符条件的文件中查找符合正则表达条件的字符串。若单引号内容，即看到即所得；而双引号内容先经转换，例如其中命令，变量等，然后搜索

-v, 反向搜索

-r, 递归搜索，同时处理多层子目录里的文件

-q, 静默输出，不输出人结果，用于获得return value

-i, 忽略大小写

-w, 整词比对，类似<word>\

-n, 同时输出行号

-c, 只输出符合比对的行数

-l, 只输出符合比对的文件名称

在当前目录后缀有“txt”字样的文件中查找含有test字符的文件，并打印该行

```
grep test *txt
```

以递归方式查找目录下所有符合条件的文件，例如/share/Data/目录及其子目录下所有包含"update"字符串的文件，并打印该行

```
grep -r update /share/Data
```

反向查询文件名包含diamond的文件中不含jewel的行

```
grep -v jewel *diamond*
```

- find

find在指定目录下查找符合通配符条件的文件，根据path和expression查找文件

```
find path [options] expression [-print] [-exec/-ok command] {} \;
```

-amin n, 在过去n分钟读取过的文件，+/-分别表示之前或之内

-atime n, 在过去n天被读取过的文件，+/-分别表示之前或之内

-cmin n, 在过去n分钟状态被修改过的文件，+/-分别表示之前或之内

-ctime n, 在过去n天状态被修改过的文件，+/-分别表示之前或之内

-mmin, n, 在过去n分钟修改过的文件，+/-分别表示之前或之内

-mtime n, 在过去n天修改过的文件，+/-分别表示之前或之内

-path p/ipath p, 路径符合p的文件，ipath忽略大小写

-name name/-iname name, 文件名符合name的文件，iname忽略大小写

-type type, 文件类型type, d表目录, f表一般文件, l表符号连接

-perm xxx, 查找指定权限的文件

-size xxx, 查找文件大小xxx的文件, k表kilobytes, b表bytes, +/-分别表示大于/小于

-maxdepth n, 限定查询目录层数n

-print0, 不打印结尾的换行

-ok, 执行前增加询问

列出当前目录及其子目录下所有最近20天更新过的文件

```
find $(pwd) -ctime -20
```

查找指定目录下更改时间在7天前的普通文件, 并在删除前询问

```
find /your/dir -type f -mtime +7 -ok rm {} \;
```

- xargs

xargs为命令传递的过滤器, 用于组合多个命名, 默认命令为echo, 这意味着通过管道传递给xargs的输入将会包含换行和空白, 通过xargs处理, 换行和空白将会被空格取代。

```
somecommand | xargs -item command
```

-n num, 表命令在执行时一次使用的argument个数, 默认所有

```
$for loop in {a..z}; do echo $loop; done | xargs -n5
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
```

-l {}, 使用替代字符{}, 当-l和xargs结合使用, 每个参数命令均被{}替代, 同时执行一次

```
$for loop in {a..z}; do echo $loop; done | xargs -l {} echo {}, "once a letter"
a,once a letter
b,once a letter
c,once a letter
d,once a letter
```

- split

```
split [-a suffix_length] [-b byte_count[k|m]] [-l line_count] [-p pattern][file
[name]]
```

split将文件分割成数个, 默认情况下按照每1000行进行分割

-l n, 指定行数n分割

-b n, 指定每部分输出文件大小, 默认为byte, 可使用k表kilo bytes, m为1024k

-a n, 指定分割后文件名后缀长度

```
$split -b 10m NUT_DATA.txt NUT_DATA_
carlos@huizhendeMacBook-Pro 21:25:54 /Scripts_practice/python_scripts
$l
total 172560
-rw-r--r--  1 carlos  wheel   5.4M Jun 17 21:25 NUT_DATA_ad
-rw-r--r--  1 carlos  wheel   10M Jun 17 21:25 NUT_DATA_ac
-rw-r--r--  1 carlos  wheel   10M Jun 17 21:25 NUT_DATA_ab
-rw-r--r--  1 carlos  wheel   10M Jun 17 21:25 NUT_DATA_aa
```

- paste

paste用于文件合并

```
paste [-s] [-d list] file ...
```

默认将多个文件合并，输出STDOUT

-s, 将文件多行合并成一行显示

```
$echo {a..z} | xargs -n10
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
carlos@huizhendeMacBook-Pro 21:32:28 /Scripts_practice/python_scripts
$echo {a..z} | xargs -n10 | paste -s -
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
```

-d, 指定输出间隔字符

```
$echo {a..z} | xargs -n10 | paste -s -d " " -
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
carlos@huizhendeMacBook-Pro 21:34:31 /Scripts_practice/python_scripts
$echo {a..z} | xargs -n10 | paste -s -d "\n" -
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
```

- cut

cut从文件的每一行剪切字节，字符和字段写至标准输出

-b n, 指定字节为单位分割

-c n, 指定字符为单位分割

```
$printf "%s" {a..z}|cut -b 3
c
carlos@huizhendeMacBook-Pro 21:44:03 /Scripts_practice/python_scripts
$printf "%s" {a..z}|cut -c 3
c
```

-d, 定义分割符，默认制表符

-f, 与-d一起使用，指定输出区域

```
$echo {a..z}|cut -d " " -f 3
c
carlos@huizhendeMacBook-Pro 21:46:02 /Scripts_practice/python_scripts
$echo {a..z}|cut -d " " -f 26
z
```

- sort

sort用于将文本文件进行排序，默认以行为单位排序

```
sort [-bcCdfghiRMmnrsvz] [-k field1[,field2]] [-S memsize] [-T dir] [-t char] [-o
output] [file ...]
```

- b, 忽略每行前面出现的空格字符
- c, 检查文件是否已经按照顺序排序
- f, 排序时，将小写字母当作大写字母，即忽略大小写
- r, 以相反顺序进行排序
- t, 指定排序时所有栏位分割字符
- k, 指定用于排序栏位
- d, 排序时，处理英文字母，数字及空格，忽略其他字符
- u, 排序过程中去除重复行
- o, 指定输出，可使用-o将排序后结果输出原文，使用重定向将为空
- n, 按照数字大小排序

指定域排序，需要使用逗号，限制域区间，否则会延伸至结尾，同时使用小数点指定域的字符

```
$sort -t: -k1,1r -k2,2n test.sh
pear:80:2.3
pear:90:2.3
orange:20:3.4
orange:30:3.4
banana:30:5.5
apple:10:2.5
```

- uniq

检查并删除文本中出现的重复行，一般与sort命令结合使用

- c, 每行旁边显示该行重复出现的次数
- d, 仅显示重复出现的行
- u, 仅显示出现一次的行
- i, 忽略大小写

当重复行不相邻时，**uniq**命令不起作用，因此先使用**sort**，**sort file|uniq**

- fmt

fmt用于编排文本文件

-w, 指定每行字符数

```
$cat test.sh|xargs | fmt -w 35
banana:30:5.5 apple:10:2.5
pear:90:2.3 pear:80:2.3
orange:20:3.4 orange:30:3.4 test 30
test 30 test 30 Hello 95 Hello 95
Hello 95 Hello 95 Linux 85 Linux 85
```

- tr

tr用于转换或删除文件中的字符

-c, 表示反选, 符合的不处理, 不符合的转换

-d, 删除指定字符

-s, 缩减连续重复字符为指定单个字符

```
$cat test.sh|xargs |tr -d a-z
:30:5.5 :10:2.5 :90:2.3 :80:2.3 :20:3.4 :30:3.4 30 30 30 H 95 H 95 H 95 L 85 L 85
carlos@huizhendeMacBook-Pro 22:30:46 /Scripts_practice/python_scripts
$cat test.sh|xargs |tr -c a-z Z
bananaZZZZZZZappleZZZZZZZpearZZZZZZZpearZZZZZZZorangeZZZZZZZorangeZZZZZZZtestZZZZtestZZZZtestZ
ZZZZelloZZZZZelloZZZZZelloZZZZZelloZZZZZlinuxZZZZZlinuxZZZZcarlos@huizhendeMacBook-Pro 22:30:53 /Scrip
ts_practice/python_scripts
$cat test.sh|xargs |tr a-z A-Z
BANANA:30:5.5 APPLE:10:2.5 PEAR:90:2.3 PEAR:80:2.3 ORANGE:20:3.4 ORANGE:30:3.4 TEST 30 TEST 30 TEST
30 HELLO 95 HELLO 95 HELLO 95 HELLO 95 LINUX 85 LINUX 85
```

- ps

ps命令用于显示当前进程(process)状态

-A, 列出所有行程

-w, 显示加宽可以显示较多的咨询

-u, 查看指定用户

-C, 查看指定行程

-f, 查看格式化信息表

-axjf, 树状结构显示进程

-ax, 显示所有使用者的行程

a(x)输出格式:

USER PID %CPU \$MEM VSZ RSS TTY

行程拥有者; pid; 占有CPU使用率; 占有内存使用率; 占有虚拟内存大小; 占有内存大小; 终端的次要装置号码;

STAT: 该行程状态(D,无法中断的休眠状态;R,正在执行中;S,静止状态;T,暂停执行;Z,不存在在但暂时无法消除;W,没有足够的内存分页可分配;<,高优先的行程;N,低优先序的行程)

START:行程开始时间

TIME:执行的时间

COMMAND:所执行的指令

- scp

```
scp [options] file_source file_target
```

基于ssh登陆，进行具有security的文件的copy，就是把当前一个文件copy到远程另外一台主机上：

```
scp local_file remote_username@remote_ip:remote_folder
```

```
scp /home/open/tools.tar.gz root@113.223.3.23:/home/desktop
```

会提示输入远程主机的密码

把文件从远程主机copy到当前系统：

```
scp remote_username@remote_ip:remote_folder local_file
```

```
scp root@113.223.3.23:/home/desktop/tools.tar.gz /home/open/tools.tar.gz
```

-r 用于复制目录

-v 显示进度

-C 使能压缩选项

-P 选择端口

-4 强行使用IPV4地址

-6 强行使用IPV6地址

rsync -P --rsh=ssh home.tar user@192.168.205.34:/home/user/home.tar 断点续传: rsync -P --rsh=ssh

-P 包含了'--partial-progress'，部分传送和显示进度

--rsh=ssh 表示使用ssh协议传送数据
