

Linux Shell整理

2019 年 3 月 5 日

1 常见基本命令

格式

```
command [option] [param1] [param2] ...
```

断行 当命令过长，一行写不开，可以使用“{}”符号将命令分成两行。

1.1 简单命令

who 列出所有使用系统的用户信息

echo 输出后续命令：有单引号‘’则原样输出，无单引号则以字符串格式输出（忽略字间连续空格）

date 显示当前日期

cal 显示日历

clear 清屏

passwd 修改当前用户密码

1.2 文件相关命令

1.2.1 文件显示

cat 显示文件内容

more 显示文件内容，但是每次只显示一屏

less 显示文件内容，可上下翻页

head 显示文件内容，但只显示前几行内容

tail 显示文件内容，但只显示后几行内容

touch 刷新修改时间

1.2.2 匹配, 筛选, 排序指定内容

grep 筛选含有指定内容的词或短语

```
//在文件f1和f2中寻找含有main或printf的行  
grep -i 'main|printf' f1 f2
```

sort ?

uniq 删除重复行

1.2.3 文件的比较

comm 比较两份文件, 输出1) 仅在文件1中的行; 2) 仅在文件2中出现的行; 3) 两个文件的共同行。要求两份文件事先排序

diff 比较两个文件。不要求两份文件事先排序

uniq 删除重复行

1.2.4 复制, 删除和移动

cp 复制文件

rm 删除文件

mv 移动文件或修改名称

1.2.5 统计文件信息

wc 统计文件的字数、行数等信息。

1.3 目录相关命令

1.3.1 新建或删除目录

mkdir 新建目录

rmdir 删除目录

1.3.2 切换工作目录和显示目录内容

cd 切换工作目录

ls 显示当前目录下目录

pwd 显示当前工作目录

1.3.3 目录的连接

ln 硬连接

ln -s 软连接

1.4 用户和权限

chmod 修改权限

```
// file权限, 新增all用户的可执行x权限
chmod a+x file
// file权限, user用户的权限改为只读r
chmod u=r file
```

umask 新建文件的默认权限, 删除对应权限

```
// user权限不取消, group权限取消w权限, other权限都取消
umask u=,g=w,o=rwxg
```

chgrp 改变文件的所属用户组

chown 改变文件所属的用户或组

1.5 进程相关命令

ps 显示当前系统中所有进程

kill 终止某进程

sleep 暂停某进程

1.6 解压操作

《解压专题》

2 Shell程序

这里介绍shell的编写方法及相关内容。

2.1 Shell执行方法

- 1) `bash < 脚本`
- 2) `脚本 [参数]`
- 3) `./脚本`

2.2 Shell中的特殊字符

通配符

`*` 可以用于匹配任意长度的任意字符。

`f* = f, fa, f1, fa2, facsdcs`

`?` 可以匹配一个的任意字符。

`f? = fb, fa, f1, f2, fd`

`[]` 可以匹配括号中任意一个字符。

`f[1-9] = f1, f2, f3, f4, f5`

`f[abcd] = fa, fb, fc, fd`

`f[1-4] = f[1234]` // 可以用连字符省略数字或字符

`!` 表示不匹配后续字符

`f[!1-9] = 排除 f1, f2, f3, f4, f5...`

`f[!abcd] = 排除 fa, fb, fc, fd`

模式表达式 模式表表示需要匹配的内容，如果需要多个匹配内容使“或”运算符分隔。

***(模式表)** 匹配模式表中出现的模式，可连续出现，且含不出现模式的情况。

```
file*(.c|.o) = filexx.c fileddf.o file filexx.c.o filexx.o.c
```

+(模式表) 匹配模式表中出现的模式，可连续出现，不含不出现模式的情况。

```
file+(.c|.o) = filexxxx.c filedafdf.o filexx.c.o filexxx.o.c
```

?(模式表) 匹配模式表中出现的模式，不可连续出现，且含不出现模式的情况。

```
file?(.c|.o) = file file.c file.o
```

@(模式表) 匹配模式表中出现的模式，不可连续出现，但不含不出现模式的情况。

```
file@(c|.o) = file.c file.o
```

!(模式表) 除模式表中内容，其他均匹配。

引号

“双引号” 除\$符号，‘（倒引号）符号，反斜杠符号外，其他作为普通字符。\$符号表示变量替换，可以将变量（如“\$HOME”）替换成对应的值（如“/home/user/”）。‘表示指令替换，将“中指令的返回值，替换进字符，如‘pwd’，会替换成当前文件路径。反斜杠表示转义字符。

‘单引号’ 单引号中内容将除去所有特殊效果，仅作为字符。

‘倒引号’ 如上文所述，倒引号会被替换成命令的返回值。

输入、输出重定向符

- < 将默认的输入，重定向至本文件。
- > 将默认的输出，重定向至本文件。
- >> 将默认的输出，重定向至本文件，以追加方式，附加到文件尾部。
- <<

其他

注释 用#号后接注释

管道 用|隔开，前一文件的结果作为后一命令的输入

后台运行 命令后接&符号

可以每条命令占一行；也可以将命令后接分号，写在一行。

&&与符号，表示前一条命令若执行成功，再执行后一条命令。||或符号，表示前一条命令若执行不成功，再执行后一条命令。

数组：数组名=（值1 值2 值3 ...），可以使用*和@号表示数组中所有下标。

```
A = (this is an example)
A[0] = this
A[1] = is
A[2] = an
A[*] = this is an example
A[@] = "this" "is" "an" "example"
```

读取用户输入输出命令

read

echo

2.3 控制结构

```
if 条件
then 表达式
else 表达式
fi
```

```
if 条件
then 表达式
elif 条件
then 表达式
else 表达式
fi
```