Linux Bash Shell Basics

数 据 嗨 客

Linux Shell是一个类似于传统Unix命令行界面的命令解释器。 用户可通过在命令行输入文本命令或创建包含多个命令的脚本文件来控制计算机的运行。用户通常使用终端模拟器与Linux Shell进行交互。

尖括号“<>”的部分需用户自行填写，例如 cat <file> 表示 cat file.txt，cd <folder> 表示 cd Documents/folder1/，以此类推.

终端快捷键

CTRL+L | 清屏.
SHIFT+Page Up/Down | 向上/下.
CTRL+A | 将光标移到行的最开始.
CTRL+E | 将光标移到行的末尾.
TAB | 自动补全代码或路径.
CTRL+R | 反序搜索历史记录.
!! | 执行上一次的指令.
CTRL+Z | 终止现在的指令.
COMMAND/ALT + -/+ | 放大或缩小显示字体.
CTRL+U | 剪切光标左侧代码.
CTRL+K | 剪切光标右侧代码.
CTRL+W | 剪切光标左侧单词.
CTRL+Y | （在 CTRL U,K或W之后）粘贴代码.

基本指令

who, whoami | 查看已登录用户/当前用户.
pwd | 查看当前路径.
q | 退出当前窗口.

ls <opt> <file | folder> | 列出指定文件下的内容，默认为当前文件夹.
其中<opt>的常用值有：
-a | 列出所有文件，含隐藏文件.
-l | 以长格式列出所有文件，字节格式.
-lh | 以长格式列出所有文件，正常格式.
-la |以长格式列出所有文件，含隐藏文件.
-lh *.png | 列出所有png文件的信息.
-lh <file> | 列出某个文件的信息.
-R | 递归式列出所有子文件.
ls -S | 按文件大小降序排序.

cd / | 前往根目录.
cd | 前往家目录.
cd .. | 回到上一级目录.
cd - | 回到之前目录.
cd <folder> | 前往指定文件夹（如文件夹名有空格，需用引号或转译字符‘\’链接）.
pwd | 显示当前文件夹.
mkdir <folder> | 创建文件夹.
du -h | 查看当前文件夹下所有文件夹的大小.
du -ah | 查看当前文件夹下所有文件的大小.
man <cmd> | 显示某个命令的帮助文档.

文件操作

touch <file> | 创建或更新文件.
cat <file> | 一次性查看文件全部内容.
less | 以标准输出形式逐页查看文件内容.
echo <obj> | 以标准输出形式打印对象.
echo $? | 输出上一个命令执行后的退出值.
head -<n> <file> | 查看文件前n行内容.
tail -<n> <file> | 查看文件后n行内容.
tail -f -<n> <file> | 实时查看文件后n行内容.
cp <Oldfile> <Newfile> | 创建文件副本.
cp <Oldfile> <folder> | 复制文件到指定文件夹.
cp -R <Oldfolder> <Newfolder> | 复制文件夹到指定文件夹.
cp -v | 在复制过程中显示进行.
mv <file> <folder> | 移动文件到文件夹.
mv <folder> <folder> | 移动文件夹到新文件夹.
mv <Oldfile> <Newfile> | 重命名文件.
mv <folder> ../ | 将文件夹向上移动一级.
rm <file> | 删除文件.
rm -i <file> | 带有确认提示地删除文件.
rm -f <file> | 强制删除文件.
rm -r <folder> | 递归式删除文件夹.
ln <file1> <file2> | 创建文件的硬链接（即文件别名）.
ln -s <file1> <file2> | 创建文件的软链接（即快捷方式）.

文件搜索

find <path> <arg> <opt> | 按条件对指定路径下的文件进行搜索.

其中<path>用来指定搜索的路径，除了指定绝对路径外，还可以使用‘.’表示当前文件夹，‘~’表示家目录，以及‘/’表示根目录.

常用<arg>和<opt>参数包括：
-name '*.png' | 搜索所有.png文件.
-iname '*.PnG' | 搜索时忽略大小写.
-amin n | 第n分钟前访问过的文件.
-cmin +n | 大于n分钟前状态改变的文件.
-mmin -n | 在n分钟内修改过的文件.
-atime n | 第n天前访问过的文件.
-ctime +n | 大于n天内状态改变的文件.
-mtime -n | 在n天内修改过的文件.
-empty | 所有空文件和文件夹.
-size +100M | 大于100M的文件
-maxdepth 3 | 查找时下至最多3级.
-mindepth 8 | 查找时从下至8级开始.
-uid 100 | 用户ID为100的文件.
-gid 100 | 组ID为100的文件.
-group staff | 属于staff组的文件.
-user penglu | 属于用户penglu的文件.
-type f | 只搜索文件类型.
-perm 755 | 搜索权限为755的文件.
<arg> ! <arg> | 满足两个条件的文件.
<arg> -o <arg> | 满足任意条件的文件.

<exp> 的其余参数需根据搜索内容的不同而定，可使用man find命令查阅find的帮助文档.

文件压缩与解压

tar命令用于将文件和目录集合转换为高度压缩的文档文件，以便于可以轻松地将文件从一个磁盘移动到另一个磁盘. 其语法为：
tar <opt> <file | dir>
其中<opt>的常用值有：
-c | 创建文档文件.
-t | 显示文档文件内容.
-x | 解压文件.
-f <NewFile> | 指定文件名称.
-z | 使用gzip压缩.
-j | 使用bzip2压缩.
-v | 显示过程.

应用举例：
tar -cvf tarfile.tar mydoc
将mydoc文件夹转换成tarfile.tar文档文件.
tar -xvf tarfile.tar
解压tarfile.tar文件到当前文件夹.
tar -tvf tarfile.tar
显示tarfile.tar文件内容.

进程管理

ps | 显示静态进程列表.
top | 显示动态进行列表.
pstree | 以层级结构显示进程.
kill <PID> | 按进程ID终止进程.
killall <Pname>|按名称终止所有相关进程.
pgrep <Pname> | 通过进程名查找进程的ID.

文本操作

grep <opt> <ptn> <file> | 在指定条件下查找文件中所包含的字符串模板，默认情况下该函数将输出所有匹配的行.
其中<opt>的常用值有：

Chen Penglu

Linux Bash Shell Basics

数 据 嗨 客

-i | 搜索时忽略大小写.
-r | 递归式搜索.
-A | 显示匹配行之后的内容.
-B | 显示匹配行之前的内容.
-C | 显示匹配行之前和之后的内容.
-h | 输出结果时不显示文件名.
-w | 只显示全字符合匹配.
-c | 计算符合模板的行数.
-n | 显示符合模板的行编号.
-v | 搜索不包含此模板的行.
-E | 以拓展正则表达式进行搜索.
-s | 不显示错误信息.
-H | 显示匹配行所属文件的文件名.
-q | 不输出任何信息.

sed 's/<ptn>/<replace>/' <file> | 流编辑器函数，用来以标准输出形式修改符合字符串模板的文字，例如：

```
echo 'hi there' | sed 's/hi/hello/'
```

wc <opt> <file> | 字数统计函数.
其中<opt>的常用值有：
-c | 显示文件的字节数.
-l | 显示文件的行数.
-L | 显示最长行的长度.
-m | 显示文件中的字符数.
-w | 显示文件的字数.

cut <opt> <file> | 切片函数，用来以标准输出形式切片字符串.
其中<opt>的常用值有：
-b | 只选中指定的这些字节.
-c | 只选中指定的这些字符.
-d | 自定义字段分隔符，默认为“TAB”.
-f | 显示指定字段的内容.
-n | 与“-b”连用，不分割多字节字符.

awk <opt> '{<cmd> <opt>}' <file> | 高级文本分析函数，它逐行读取输入流中的内容并对每一行执行指定命令，例如：

```
awk -F, '{print $1,$5}' data.csv
```

即：使用逗号来分割每一行并把每一行的第一和第五个字段打印出来.

sort <opt> <file> | 将指定文件内容排序.
其中<opt>的常用值有：
-b | 忽略每行前面的空格符.
-f | 排序时忽略字符大小写.
-n | 按数值来排序，避免10比2小的情况.
-o | 重定向输出结果来写入文件.
-r | 降序排列（默认为升序）.
-t | 指定列的分隔符.
-u |输出时去除重复行.
-R | 随机排序.

SHELL 编程

Shell脚本就是一个包含Linux命令的可执行文本文件，在该文件中可以放置任何需要的命令.

条件比较

表达式	意义
[-z STRING]	空字符
[-n STRING]	非空字符
[NUM -eq NUM]	等于
[NUM -ne NUM]	不等于
[NUM -lt NUM]	小于
[NUM -le NUM]	小于或等于
[NUM -gt NUM]	大于
[NUM -ge NUM]	大于或等于
[! EXPR]	Not
[X] && [Y]	And
[X] || [Y]	Or

缩写表达

表达式	意义
{A,B}	等于A B
{A,B}.js	等于A.js B.js
{1..5}	等于1 2 3 4 5

特殊变量

符号	意义
name=value	将值赋值给变量
#!/bin/bash	指定脚本运行所需的Shell类型
$0	查看当前脚本的文件名
$1...$9	传递给脚本命令行的某个参数
$#	传递给脚本命令行的参数个数
$*	传递给脚本命令行的所有参数
$$	查看当前Shell的进程ID
$?	查看上个命令的退出状态

文件权限

sudo <cmd> | 以任何人身份运行代码，默认为系统管理员root.
sudo -u penglu <cmd> | 以用户penglu身份运行代码.
sudo !! | 以管理员身份重新运行上一步代码.

ls -l <file> | 可用来查看文件权限，第一位字符代表文件类型：其中-代表一般文件，d代表文件夹，l表示链接，后面的9位字符三个为一组，分别代表所有者权限、组员权限以及其它用户的权限.

chown <owner> <file> | 更改文件所有者.
chown <owner>:<group> <file> | 更改文件所有者和所属组.
chown -R 递归式更改所有子文件从属关系
chmod <obj><act><perm> <file> | 对指定文件的目标对象修改权限，例如：

```
chmod ug+rw file.txt.
```

对象包括： 所有者u，所属组g，和其它人o.
动作包括：增加权限+，取消权限-，设定权限=.
权限包括：读取r，写入w，执行x.
权限也可以用数字0或1表示，例如rwx等于111、r--等于100。如果将这些二进制数再次转换为十进制，就可以使用简写来修改权限，例如：

```
chmod 660 file.txt.
```

VIM 编辑器基础

vim <file> | 用VIM编辑器查看文件. 在VIM编辑器中可使用以下命令对文件进行查看和编辑：
h, j, k, l | 向左/下/上/右移动光标.
w | 移动到下一个单词.
b, e | 移动到当前词的词首/词尾.
B, E | 移动到空格分词的词首/词尾.
0, $ | 移动到当前行的行首/行尾.
12G, :12 | 移动到文件第12行.
gg, G | 移动到文件开始/结尾.
H, M, L | 移动到屏幕的上方/中间/下方.
(), { }| 移动到上/下一句，上/下一段.
X, x, D | 删除光标左/右/至末尾的字符.
dd, :d | 删除当前行.
/string, ?string | 向下/上文搜索.
n/N | 查找上一个/下一个.
u, Ctrl+r, U | 撤销/重做/撤销整行.
:s/strA/strB/g |将所有字符串A换成B.
r, dd, yy, p, P | 替换/剪切整行/复制整行/粘贴在光标前/粘贴在光标后.

特殊字符

> | 重定向输出 .
< | 重定向输入.
* | 匹配0或多个字符.
? | 匹配任意一个字符.
[...] | 匹配指定字符中的一个.
| | 管道命令.
$ | 访问变量中的值.
\| 转意字符.
~ | 使用命令的输出内容.
‘ ’ | 移除所有特殊符号的意义.
“ ” | 移除除$\外其余特殊符号意义.
<cmd1> && <cmd2> | 当命令1成功后执行2.
<cmd1> || <cmd2> | 当命令1出错后执行2.
<cmd> & | 后台运行命令.

Chen Penglu
数 据 嗨 客 出 品