

Linux - 目录树

1. / 系统根目录
2. /bin 包含系统启动和运行所需要的二进制文件(binary)
3. /dev 设备节点特殊目录。内核将它能够识别的所有设备存放在这个目录(linux一切皆文件 设备也要看成文件)
4. /etc 一般的配置文件都在里面
5. /home 由于Linux是多用户的 所以每个用户都有自己独立空间,普通的用户只有权限在这里创建文件。
6. lib linux 核心系统使用的共享库文件
7. /media 可移除媒体设备挂载点。比如usb驱动 cd-rom等。当设备插入计算机后,会自动的挂载到这个目录节点下面
8. /mnt 早期linux系统中 /mnt 目录包含手动挂载可移除的设备挂载点
9. /opt 安装可选软件,一般用户自己下载的软件放在这里
10. /root root账户的主目录
11. /sbin 系统二进制文件。重要的系统文件。为超级用户预留的
12. /tmp 提供用户存放各种临时系统文件的目录 可以配置每次重启系统时清空这个目录
13. /usr 包含普通用户使用的所有程序和相关文件
14. /usr/bin 目录中放置了一些Linux发行版安装的可以执行程序。存储成千上万个程序
15. /usr/lib /usr/bin中使用的共享库
16. /usr/local 由源代码编译好的程序通常安装在/usr/local/bin中
17. /usr/sbin 更多的管理员程序
18. /var 存放有可能改变的目录,比如说系统日志

Linux基础shell命令 – 文件和目录管理 – ls

文件和目录的管理：

ls : 查看文件或者目录 list

ls [-aAdfFhIlNrRDt] 文件名或目录名

ls

ls -a => 列出全部的文件，连同隐藏文件（开头为.的文件）

ls -d => 仅列出目录本身，而不是列出目录内的文件数据

ls -l => 不仅列出文件名，还包含文件的属性与权限等数据

ls -a /usr/bin

ls -d /usr/bin

ls -l /usr/bin

ls -al

ls -al /usr/bin

-a	-all 列出目录下的所有文件，包括以.开头的隐含文件
-l	除了文件名之外，还将文件的权限、所有者、文件大小等信息详细列出来
-d	-directory 将目录象文件一样显示，而不是显示其下的文件
-h	-human-readable 以容易理解的格式列出文件大小 (例如 1K 234M 2G)
-t	以文件修改时间排序

Linux基础shell命令 – 文件和目录管理 – cp

```
cp : 复制文件或者目录 copy
cp ~/.bashrc /tmp/bashrc => 用root的身份，将主文件夹下的.bashrc复制到/tmp下，并更名为bashrc
ls -al /tmp/bashrc
cp ~/.bashrc /tmp/bashrc
cp -i ~/.bashrc /tmp/bashrc =>若目标文件已经存在时，在覆盖的时候会先询问动作的进行
cd /tmp
cp /var/log/wtmp .
ls -l /var/log/wtmp wtmp => 两个文件的属性和权限不一样
cp -a /var/log/wtmp wtmp_2
ls /var/log/wtmp wtmp_2 => 两个文件的属性和权限一样
cp /etc/ /tmp =>如果是复制目录，则不能直接复制，需要加上-r参数
cp -r /etc/ /tmp
```

-a	复制文件和目录，以及它们的属性，包括所有权和权限。默认情况下，复本具有用户所操作文件的默认属性。
-i	在重写已存在文件之前，提示用户确认。如果这个选项不指定，cp 命令会默认重写文件。
-r	递归地复制目录及目录中的内容。当复制目录时，需要这个选项（或者-a 选项）。
-u	当把文件从一个目录复制到另一个目录时，仅复制 目标目录中不存在的文件，或者是文件内容新于目标目录中已经存在的文件。
-v	显示翔实的命令操作信息
-n	不要覆盖已存在的文件(使前面的 -i 选项失效)

Linux基础shell命令 – 文件和目录管理 – mv rm

```
mv 移动文件与目录，或者更名 move
cd /tmp
cp ~/.bashrc bashrc
mkdir mvtest
mv bashrc mvtest
mv mvtest mvtest1
cp ~/.bashrc bashrc1
cp ~/.bashrc bashrc2
mv bashrc bashrc1 bashrc2
```

-i	在重写已存在文件之前，提示用户确认。如果这个选项不指定， cp 命令会默认重写文件。
-u	当把文件从一个目录移动到另一个目录时，仅移动目标目录中不存在的文件，或者是文件内容新于目标目录中已经存在的文件。
-v	显示翔实的命令操作信息
-f	如果目标文件已经存在，不会询问而直接覆盖

```
rm 移除文件或者目录 remove
rm [-fir] 文件或目录
cd /tmp
rm -i bashrc
rm -i bashrc*
rm /tmp/etc/
rm -r /tmp/etc/
rm -rf /tmp/etc/
```

-i	在删除之前，提示用户确认。
-r	递归地删除目录及目录中的内容
-f	忽视不存在的文件，不显示提示信息
-v	显示翔实的命令操作信息

Linux基础shell命令 – 显示命令

1. `type` - 说明怎样解释一个命令名
2. `which` - 显示会执行哪个可执行程序 定位
 - 在 `PATH` 变量指定的路径中搜索可执行文件的所在位置。它一般用来确认系统中是否安装了指定的软件。
3. `whereis` - 主要用于定位可执行文件、源代码文件、帮助文件在文件系统中的位置
 - `whereis` 命令查找速度非常快，这是因为它根本不是在磁盘中漫无目的乱找，而是在一个数据库中 (`/var/lib/mlocate/`) 查询。这个数据库是Linux 系统自动创建的，包含有本地所有文件的信息，并且每天通过自动执行 `updatedb` 命令更新一次。也正是因为这个数据库要每天才更新一次，就会使得 `whereis` 命令的搜索结果有时候会不准确
4. `help` - 帮助文档
5. `man` - 显示命令手册页
6. `info` - 显示命令 `info`
7. `alias` - 创建命令别名

Linux基础shell命令 – 重定向配合使用的命令-1

cat - 连接文件

查看文件内容:

cat 由第一行开始显示文件内容

cat /tmp/testtouch

cat -n /tmp/testtouch

tac 将 cat 反写过来, 所以它的功能就跟 cat 相反, cat 是由第一行到最后一行连续显示在屏幕上, 而 tac 则是由最后一行到第一行反向在屏幕上显示出来

xargs - 把输出的命令改成一行为自己的指定行数打印

-n 每行打印多少个

less - 分页显示

less 一页一页翻动

less /etc/man_db.conf

空白键 : 向下翻动一页;

[pagedown] : 向下翻动一页;

[pageup] : 向上翻动一页;

/字符串 : 向下搜寻“字符串”的功能;

?字符串 : 向上搜寻“字符串”的功能;

n : 重复前一个搜寻 (与 / 或 ? 有关!)

N : 反向的重复前一个搜寻 (与 / 或 ? 有关!)

g : 前进到这个数据的第一行去;

G : 前进到这个数据的最后一行去 (注意大小写);

q : 离开 less 这个程序;

Linux基础shell命令 – 重定向配合使用的命令-2

- sort – 排序文本行
- grep – 用来找到文件中的匹配文本
`grep 'root' /etc/passwd`
- head – 输出文件开头部分
- tail – 输出文件结尾部分
`tail -f /var/log/system.log` //实时监听文件最新内容

Linux基础shell命令 – 连接-（软连接和硬连接）

在linux中[ln]这个命令用来创建连接文件。

共有两种连接文件：一种是类似与Windows的快捷方式(软链接)，另一种就是通过文件系统的inode来产生的新的文件名(硬链接)。

◇ 硬链接

由于硬链接有不可以垮文件系统，不能为目录创建等限制，因此使用较少。不论修改哪个文档，内容都会一起变。因为他们都是指向同一个block。但是当你删掉了其中一个文档的时候，是对另一个没有影响。

◇ 软链接

这个就类似于Windows的快捷方式，使用较多。

使用方式：ln [-sf] 源文件 目标文件

选项：-s 就表示创建软链接了，不加的话就是创建硬链接

-f 表示存在的时候删除在创建

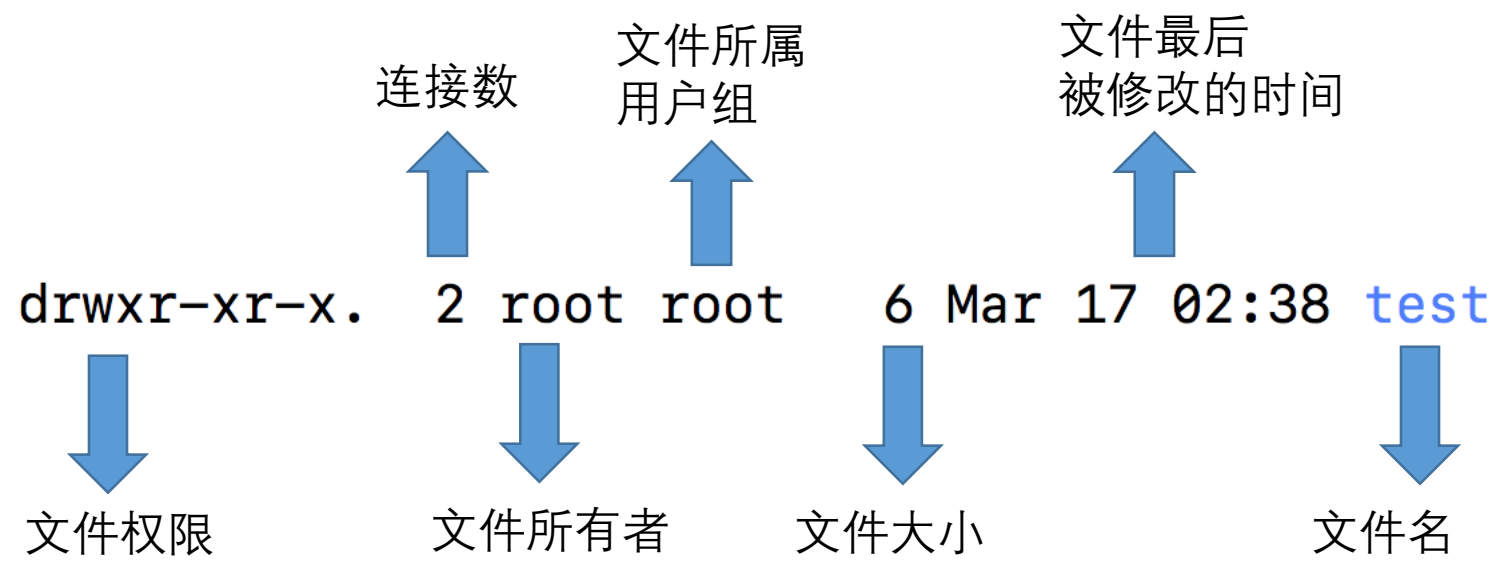
Linux基础知识 – 用户(User 和 Group)

■ 新增用户: `useradd user1` 新增user1用户
 `passwd user1` 给用户user1设置密码

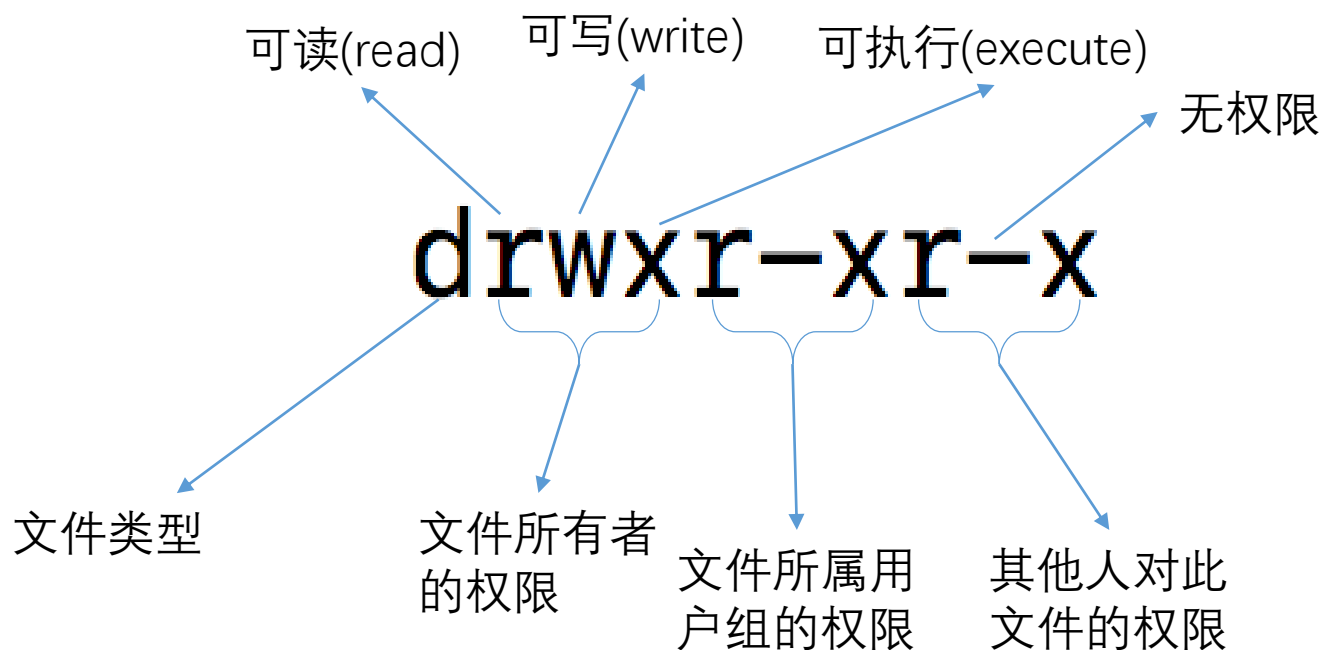
新创建的用户会在/home下创建一个用户目录user1

- 1、用user1用户ssh登陆到虚拟机
- 2、`id user1`, 可以看出一个user属于一个默认的用户组
- 3、`groupadd group1`
- 4、`useradd user2 -g group1`
- 5、`passwd user2`
- 6、`usermod -g group1 user1`
- 7、`groupadd group2`
`useradd user3 -g group2`
`passwd user3`

Linux基础知识 – 文件权限



Linux基础知识 – 文件权限



第一个字符代表这个文件的类型：

d 则表示目录

- 则表示文件

l 则表示连接文件

b 则表示设备文件里面可供存储的接口设备

c 则表示设备文件里面的串行端口设备，例如键盘、鼠标等

Linux基础知识 – 改变文件属性

chgrp - change group

chown - change owner

chgrp [-R] 用户组 dirname/filename

-R表示进行递归的持续更改，连同子目录下的所有文件、目录

chown [-R] 用户名 文件或目录

chown [-R] 用户名:用户组名 文件或者目录

-R表示进行递归的持续更改，连同子目录下的所有文件、目录

常见的例子为： 复制文件给用户之外的其他人时，需要用到chgrp和chown

Linux基础知识 – 改变文件权限

chmod

1、数字类型改变文件权限

r=4
w=2 文件的权限为: [-rwxrwx---]
x=1

则: owner = rwx = 4 + 2 + 1 = 7

则: group = rwx = 4 + 2 + 1 = 7

则: others = --- = 0 + 0 + 0 = 0

所以该文件的权限就是770

2、符号类型改变文件权限

chmod	u			
	g	+(加入)	r	文件或目录
	o	-(除去)	w	
	a	=(设置)	x	

chmod u=rwx,g=rx,o=r filename

Linux基础知识 – 权限实践

实践一：

在root账号下创建一个空文件：

```
touch /tmp/test.txt
```

```
ls -al /tmp/test.txt
```

```
-rw-r--r--. 1 root root 0 3月 18 14:45 /tmp/test.txt
```

```
vi /tmp/test.txt
```

在user1账号下读写/tmp/test.txt =>有读的权限，没有写的权限

在root账号下对/tmp/test.txt增加others的写的权限

```
chmod 646 /tmp/test.txt
```

在user1账号下读写/tmp/test.txt =>有读的权限，有写的权限

实践二

现在只想将/tmp/test.txt的读写权限给用户1和用户2

将/tmp/test.txt的group改成group1

```
chgrp group1 /tmp/test.txt
```

将/tmp/test.txt的权限改成：

```
chmod 660 /tmp/test.txt
```

将user3所属的group改成group1：

```
usermod -g group1 user3
```

```
su - user1
```

```
vi /tmp/test.txt
```

```
exit
```

```
su - user3
```

```
vi /tmp/test.txt
```

```
exit
```

Linux基础知识 – vi-移动和翻页命令

[h]或[←]: 光标向左移动一个字符

[j]或[↓]: 光标向下移动一个字符

[k]或[↑]: 光标向上移动一个字符

[l]或[→]: 光标向右移动一个字符

但是一个一个的移动貌似有点效率非常的低, 这时候我们可以使用【数字加命令】的方式来移动。

比如: 30j: 向下移动30行 10→: 向右移动10个字符

[Ctrl] + [f]: 屏幕向下移动一页, 相当于[Page Down]

[Ctrl] + [b]: 屏幕上移动一页, 相当于[Page UP]

[Ctrl] + [d]: 屏幕向下移动半页

[Ctrl] + [u]: 屏幕向上移动半页

[+]: 光标移动到非空格符的下一列

[-]: 光标移动到非空格符的上一列

n<space>: n代表数字, 按下数字后, 在按空格, 光标会向右移动到这一行的n个字符。

0或[home]: 这个是数字0, 表示移动到这一行的最前面字符处。

\$或[End]: 移动到这一行的最后面字符处。

[H]: 光标移动到这个屏幕的最上方那一行的第一个字符。

[M]: 光标移动到这个屏幕的中央那一行的第一个字符

[L]: 光标移动到这个屏幕的最下方那一行的第一个字符。

[G]: 移动光标到这个文件的最后一行。

[nG]: 你表示数字, 移动到这个文档的第n行

[gg]: 移动到这个当好的第一行, 相当于[1G]

[n<Enter>]: n表示数字, 光标向下移动n行

Linux基础知识 – vi-检索与替换命令

[/word]: 向光标之下寻找一个名称为word的字符串

[?word]: 向光标之前寻找一个名称为word的字符串

[n]: 这个n是英文按键，代表重复前一个搜寻的动作，假如说你用[/name]来搜寻name之后，可以按[n]来继续查找下一个

[N]: 这个和[n]相反，是用来查找上一个

[:n1,n2s/word1/word2/g]: n1和n2为两个数字，表示在n1和n2行之间寻找word1这个字符串，并将该字符串替换为word2

[:1,\$s/word1/word2/g]: 从第一行到最后一行寻找word1字符串，并替换为word2

[:1,\$s/word1/word2/gc]: 从第一行到最后一行寻找word1字符串，并替换为word2，且在替换之前显示提示字符，让用户来确认是否要替换。

Linux基础知识 – vi-删除与复制命令

[x,X]: 在一行当中, [x]表示向后删除一个字符, 相当于[del],[X]表示向前删除一个字符, 相当于[backspace]

[nx]: n为数字, 表示向后连续删除n个字符

[nX]: n为数字, 表示向前连续删除n个字符

[dd]: 删除光标所在的那一整行。

[ndd]: n为数字, 删除光标所在的行向下n行

[d1G]: 删除光标所在的行到第一行的所有数据(1G表示第一行)

[dG]: 删除光标所在行到最后一行的所有数据

[d\$]: 删除光标所在处到改行的最后一个字符

[d0]: 这个是数字0, 删除光标所在处到该行的最前面一个字符

[yy]: 复制光标所在的那一行

[nyy]: n为数字。复制光标所在行向下n行

[y1G]: 复制光标所在行到第一行的所有数据

[yG]: 复制光标所在行到最后一行的所有数据

[y0]: 复制光标所在字符到行首的所有数据

[y\$]: 复制光标所在字符到行尾的所有数据

[p,P]: p为将已复制的数据在光标下一行粘贴上, P为贴在光标的上一行,

[J]: 将光标所在行与下一行的数据结合成一行

[c]: 重复删除多个数据, 例如删除10行[10cj]

[u]: 复原前一个动作

[Ctrl + r]: 重复上一个动作

[.]: 这是个小数点, 表示重复前一个动作

Linux基础知识 – vi- 编辑模式与命令模式

在一般模式下按下面这些键就可以进入编辑模式。

[i,I]: 进入插入模式, [i]为从光标所在处插入。[I]为在目前所在行的第一个非空格处开始插入

[a,A]: 进入插入模式, [a]为从光标所在的下一个字符处开始插入,[A]为从光标所在行的最后一个字符处开始插入。

[o,O]: 进入插入模式, [o]表示在目前光标所在的下一行处插入新的一行, [O]表示在目前光标所在处的上一行插入新的一行

[r,R]: 进入替换模式, [r]表示取代光标所在的那一个字符一次, [R]会一直取代光标所在的文字, 知道按下[Esc]为止。

[Esc]: 退出编辑模式

命令模式

[:w]: 保存, 不退出

[:w!]: 强制保存, 即使权限是只读的时候, 也写入

[:q]: 退出vi

[:q!]: 强制退出, 比如有时我们修改了文档, 又不想保存

[:wq]: 保存并退出,[:wq!]表示强制保存并退出

[ZZ]: 表似乎档案没有变动, 则不储存离开, 变动了, 则存储离开

[:w filename]: 将编辑的数据储存为另一个新的文档

[:r filename]: 在编辑的数据中, 读入另一个文档的数据, 内容加到游标所在行的后面

[:n1,n2 w filename]: 将n1到n2的内容储存到filename这个档案

[:! command]: 暂时离开vi到指令列模式下执行command的显示结果。例如: [:! ls /home] 即可在vi中查看/home地下 ls的输出结果

[:set nu]: 显示行号

[:set nonu]: 取消显示行号

Linux基础知识 – 查看进程

ps命令查找与进程相关的PID号:

ps a 显示现行终端机下的所有程序，包括其他用户的程序。

ps -A 显示所有程序。

ps c 列出程序时，显示每个程序真正的指令名称，而不包含路径，参数或常驻服务的标示。

ps -e 此参数的效果和指定"A"参数相同。

ps e 列出程序时，显示每个程序所使用的环境变量。

ps f 用ASCII字符显示树状结构，表达程序间的相互关系。

ps -H 显示树状结构，表示程序间的相互关系。

ps -N 显示所有的程序，除了执行ps指令终端机下的程序之外。

ps s 采用程序信号的格式显示程序状况。

ps S 列出程序时，包括已中断的子程序资料。

ps -t<终端机编号> 指定终端机编号，并列出于该终端机的程序的状况。

ps u 以用户为主的格式来显示程序状况。

ps x 显示所有程序，不以终端机来区分。

最常用的方法是ps aux,然后再通过管道使用grep命令过滤查找特定的进程,然后再对特定的进程进行操作。

```
ps aux | grep program_filter_word,ps -ef |grep tomcat
```

ps -ef|grep java|grep -v grep 显示出所有的java进程，去处掉当前的grep进程。

Linux基础知识 – 关闭进程

kill

-s 指定发送信号

-l 根据信号名显示信号编号，若果不加信号的编号参数，则使用“-l”参数会列出全部的信号名称

-a 当处理当前进程时，不限制命令名和进程号的对应关系

-p 指定kill 命令只打印相关进程的进程号，而不发送任何信号

-u 指定用户

信号名	信号编号	信号含义
HUP	1	终端断线
INT	2	中断
QUIT	3	退出
TERM	15	终止
KILL	9	强制终止
CONT	18	继续（与STOP相反）
STOP	19	暂停

示例：

关闭apache用户下所有进程

```
kill -u apache.apache
```

强制关闭进程ID为1937的进程，参数s可以省略。

```
kill -s 9 1937
```

```
kill -9 1937
```