

Linux 第四章

1.Linux的哲学思想

1.1.一切皆为文件

在Linux系统中，不仅仅是数据以文件的形势存在，而是Linux把所有的资源，包括硬件设备都组织成了文件。比如硬盘以及硬盘中的每一个分区在Linux系统中都被视为文件。

```
vim /etc/ssh/sshd_config // ssh配置文件
ls /dev/sda // 硬盘所在文件夹
```

1.2.整个系统由众多小程序组成

在Linux系统中很少有向Windows系统中动辄上GB的大型程序，整个Linux由众多单一功能的小程序组成，每个小程序只负责实现一项具体的功能。我们之后学习的各种Linux命令几乎都是对应着一个小程序。如果要想完成一项复杂的任务，只需要将这些命令组合起来即可。

```
which ls // 查询ls程序所在位置
```

1.3.尽量避免与用户交互

Linux系统主要是操作服务器的系统，其操作方式与Windows系统有很大的区别。由于服务器管理员不能全天候的守在服务器旁，而且一个管理员往往需要管理成百上千台服务器，因为在服务器上操作最好是编写脚本来完成，从而使其自动化的完成其功能。

```
passwd ztr // 更改用户ztr的密码
echo '123' | passwd --stdin ztr // 更改用户ztr的密码--没有直接和用户进行交互
```

1.4.使用纯文本文件保存配置信息

无论Linux系统本身还是系统中的应用程序，它们的配置信息往往都是保存在一个纯文本的配置文件中。如果需要改动或者配置程序中的某项功能，只需要编辑相应的配置文件即可。

2.文件和目录的相关概念

在Linux系统中，一切皆为文件。如果要储存和管理这些文件，则需要借助这些文件所在的目录。

比如文件相当于个人，那么目录则相当于省市这些行政区划，比如：“中国/四川省/成都市/龙泉驿区/四川城市职业学院/邹堂瑞”就很明确的指出了个人。与此类似，“/etc/httpd/conf/httpd.conf”则指向了一个具体的文件。

对于文件和目录的管理是Linux系统运行维护的基本工作。

2.1.Linux的目录结构

在Windows中，为每个磁盘分区分配一个盘符，在资源管理器中通过盘符就可以访问对应的分区。每个分区使用对立的文件系统，在每个分区中都会有一个根目录，比如：C:\ 、 D:\等。

Linux系统则不同，由于Linux发行版本众多，为了同一规范，绝大多数的Linux发行版都遵循FHS(Filesystem Hierarchy Standard)文件系统层次化标准，采用同一的目录结构。按照FHS标准，整个Linux的文件系统是一个倒置的树形结构，**整个系统中只存在一个根目录，所有的目录和文件都在一个根目录下。**

在Linux系统中定位文件或目录的位置时，则用"/"进行分隔，在整个树形结构中使用"/"表示根目录，所以根目录是所有文件的起点。在根目录下按不同的特点划分了众多子目录。因此，**Linux系统的目录结构是固定的，跟磁盘分区没有任何关系。**按典型FHS目录结构划分如下：



Linux系统的目录结构由系统自动创建，每个目录都有其固定的用途。常见目录释义如下：

- /boot：存放Linux系统启动所必须的文件，Kernel便存放在这个目录里，出于系统安全的考虑，/boot目录通常被划分为独立的分区。
- /etc：存放系统和各程序的配置文件。Linux很多操作和配置都是通过修改配置文件来实现的。/etc目录类似于Windows的注册表。
- /dev：存放Linux系统的硬盘、光驱、鼠标等硬件设备。
- /bin：存放Linux常见的基本命令，任何用户都可以执行。
- /sbin:存放Linux基本的管理命令，只有管理员权限才能执行。
- /usr：软件的默认安装位置，类似于Windows的Program Files目录。
- /home：普通用户家目录(主目录)。类如普通账户"ztr"对应的家目录就是"/home/ztr"。
- /root：超级用户root的家目录。
- /mnt：一般是空的，用来临时挂载储存设备。
- /media：用户系统自动挂载可移动存储设备。
- /tmp：临时目录。用于存放系统或程序运行时产生的临时文件，可供所有用户写入操作。
- /var：存放系统运行过程中经常变化的文件，如/var/log用于存放日志文件，/var/spool/mail用于存放邮件等。
- /lib、/lib64：存放各种链接库文件。
- /proc：用于存放进程文件。
- /run：用于存放一些进程产生的临时文件，系统重启后会消失。
- /lost+found：存放系统因意外崩溃或关机时产生的碎片。

2.2.根目录和家目录

根目录只"/"下的目录，而家目录有两个：一个是超级管理员root的家目录"/root"，其他普通的用户的家目录位于/home下，我们可以通过命令添加用户，添加的用户也存放于"/home"目录下。

```
useradd 用户 // 添加用户-添加到/home目录下
```

2.3.绝对路径和相对路径

Linux系统中当我们要执行进入的是一个完整的路径的时候，那这个完整的路径就是绝对路径，如"cd /home/ztr"。当直接执行查找的文件，这个文件的目录就是相对目录，如"cd ztr"，进入相对目录的前提是你必须要在该目录的父目录才可以执行。

```
cd 目录名 // 切换进入到一个目录
ls 目录名 // 查看目录下的目录或文件
cd .. // 返回上一级目录
history // 查询历史执行命令
pwd // 查看当前所在位置
cat 文件名 // 查看文件
cd - // 回到上一个进入过的目录
```

2.4.文件和目录的操作命令

2.4.1.ls命令--查看目录内容

```
ls // 查看目录显示内容，查看文件显示目录
ls -l /etc/passwd // 查看文件的详细信息==windows属性
-rw-r--r--. 1 root root 2301 10月 29 13:33 /etc/passwd
ls -l /etc
```

释义：

第一组：文件的类别和权限。其中第一个字符代表文件的类别。"-"代表普通文件，"d"代表目录，"l"代表符号连接

"c"代表字符设备，"b"代表设备。其余6个字符代表文件的权限。c=character字符，b=block块，l=link

第二组：被硬链接的次数，文件为1，目录为2。

第三组：文件所有者。

第四组：文件所属组

第五组：文件的大小，单位为字节B-文件本身的大小，不包含下级子目录

第六组：文件被创建或最近一次修改时间

第七组：文件所属路径

```
ls -a // 显示所有文件，包含隐藏文件
"." // 表示当前目录
".." // 表示当前目录的上一级目录--父目录
```

```
[root@localhost ~]# vim test.sh
[root@localhost ~]# chmod +x test.sh
bash: chmod: 未找到命令...
相似命令是: 'chmod'
[root@localhost ~]# chmod +x test.sh
[root@localhost ~]# ls
anaconda-ks.cfg      test.sh
initial-setup-ks.cfg 公共
[root@localhost ~]# ./test.sh
Hello World!
```

书写第一个脚本

添加成可执行文件

模板 图片 下载 桌面
公共 视频 文档 音乐

执行

```
ls -ld // 显示文件本身的属性--非其文件内部列表
ls -ld -l /dev
drwxr-xr-x. 20 root root 3300 10月 29 10:43 /dev
上述命令可以简写成"ls -ld /dev"
```

```
ls -h // 人性化显示容量信息, 以K,M,G为单位表示文件大小,
h=human
```

2.4.2.touch命令--创建空文件

```
touch [参数] 文件
touch a // 创建a文件--文件名不限
touch命令不能创建目录
```

2.4.3.mkdir命令--创建目录

```
mkdir [参数] 目录名
mkdir test1 // 创建test1目录
mkdir a b c // 同时创建abc3个目录
mkdir -p x/y/z // 创建一个多级目录
```

2.4.4.rmdir命令--删除空目录

```
rmdir [参数] 目录名
rmdir test1 // 删除test1目录
rmdir -p x/y/z // 删除多级目录
```

2.4.5.rm命令--删除文件或目录

```
rm [参数] 文件或目录
rm a // 删除a文件
rm b // 删除b文件
rm -f c // 强制删除文件-不提示
rm -r c // 删除c目录
rm -rf b // 强制删除b文件-不提示
-rf参数请谨慎使用
```

2.4.6.cp命令--复制文件或目录

```
cp [参数] 原文件或目录 目标文件或目录
cp /etc/fstab /tmp/hi.txt // 将/etc/fstab文件复制到/tmp并改名为
hi.txt
cp -r // 复制目录的操作
cp -p // 复制时保留源文件的属性不变
操作:
mkdir /tmp/test // 创建目录/tmp/test
cp /etc/issue /tmp/test // 将文件/etc/issue复制到/tmp/test中
ls /tmp/test // 查看复制后新生成的文件
-----
cp -r /home /tmp/test/ // 复制目录的操作, 将/home复制
到/tmp/test/
-----
ls -ld /home/teacher/
drwx-----. 3 teacher teacher 78 10月 29 13:33 /home/teacher/
cp -r /home/teacher/ /tmp/test
ls -ld /tmp/test
drwxr-xr-x. 4 root root 46 10月 29 20:43 /tmp/test
rm -rf /tmp/test/teacher
cp -rp /home/teacher/ /tmp/test
ls -ld /tmp/test/teacher
drwx-----. 3 teacher teacher 78 10月 29 13:33 /home/teacher/
```

在执行cp命令时, 为了区分文件和目录, 建议最好在目录后面加上"/"

2.4.7.mv命令--移动文件或目录

```
mv [选项] 源文件或目录 目标文件或目录
```

需要说明的是, 如果第二个参数中的目标是一个目录, 则mv命令会将源文件移动到该目录中。如果第二个参数中的目标是一个文件, 则mv命令会将源文件进行重命名。

如果mv移动的是一个目录, 并不会和"cp"命令一样加上"-r",而是直接移动。

```
例如: 将/root/test目录中的文件test1.txt改名为test2.txt
touch /root/test/test1.txt
mv /root/test/test1.txt /root/test/test2.txt
-----
例如: 将/root/test/test2.txt移动到/tmp目录中
mv /root/test/test2.txt /tmp/
```

2.4.8.文件和目录操作的小技巧

- 用Tab键将命令或路径自动补全。如果连续按两次Tab可以列出所有的以指定字符开头的命令或路径。

```
system // 连续按两次Tab键, 列出所有的在/root目录下的system开头
的文件
```

- 用符号"!"或者组合键"Esc."(先按Esc, 再加上.)来调用上一条命令所使用的路径, 从而简化操作。

```
ls /etc/sysconfig/network-scripts/  
cd !$ // 调用上一条命令的路径
```

- 用"history"查看历史执行命令，用"!**命令编号**"再次执行指定编号的命令。

```
history // 查询历史执行命令  
!120 // 执行历史中第113条命令
```

- 通配符：通用的匹配信息符号。
 - 通配符"*"可以匹配任意数量的任意字符。
 - 通配符"?"可以在相应位置匹配任意单个字符。
 - 通配符"[]"可以匹配指定范围内任意的单个字符。

```
ls -d /etc/pa* // 显示所有的pa开头的目录和文件  
ls -d /etc/*conf* // 显示etc下所有含有conf的文件或目录  
-----  
ls -ld /dev/sd? // 以长格式列出所有以sd开头的3个字符的文件信息  
-----  
ls /dev/[df]?? // 列出dev目录下以d或者f开头并且文件名为3个字符的文件  
ls /dev/[a-c]* // 列出dev目录下以a,b,c开头的所有文件  
ls /dev/[!fhi]* // 列出dev目录下不是以f,h,i开头的所有文件
```

"*"可以匹配的字符数量没有限制，可以0个，1个或多个，而"?"和"[]"可以匹配的字符数量只能一个。

- {}扩展，{}中可以包含一个以逗号分隔的列表。并将其展开多个路径或文件名。

```
mkdir /tmp/{a,b,c} // 一次性创建/tmp/a,/tmp/b,/tmp/c三个目录  
touch /tmp/{a,b,c}.txt // 一次性创建/tmp/a.txt,/tmp/b.txt/tmp/c.txt三个文件
```

2.5.对文件内容的操作命令

Linux中的大多数文件都是文本文件，系统中提供了多个文件内容查看命令，以满足用户不同情形下查看文本内容的需求。另外，通过**grep命令**还可以在某个文本文件中找到需要的某一部分。

2.5.1.cat命令-显示文本文件内容

cat(Concatenate)命令是用来查看文本文件内容的。使用该命令时，只需要指定文件名作为参数即可。

语法：

```
cat [文本文件]
```

例如：查看/etc/redhat-release文件中的内容，获知系统的版本号。

```
cat /etc/redhat-release // 和uname -a命令相近  
// CentOS Linux release 7.9.2009 (Core)
```

cat常用的命令选项：

- **-n:显示行号，包括空白行**

例如：查看/etc/passwd中的内容，了解Linux中的用户信息

2.5.2.more和less命令-查看多行文本分屏显示

我们在执行cat命令查看文件时，往往会遇到几百上千行的文件，这是我们再去具体找哪一行就有一定的局限性，比如：

```
cat -n /etc/ssh/sshd_config
```

```
124
125 # Accept locale-related environment variables
126 AcceptEnv LANG LC_CTYPE LC_NUMERIC LC_TIME LC_
TARY LC_MESSAGES
127 AcceptEnv LC_PAPER LC_NAME LC_ADDRESS LC_TEL
MENT
128 AcceptEnv LC_IDENTIFICATION LC_ALL LANGUAGE
129 AcceptEnv XMODIFIERS
130
131 # override default of no subsystems
132 Subsystem      sftp      /usr/libexec/openssh/
133
134 # Example of overriding settings on a per-user
135 #Match User anoncvs
136 #      X11Forwarding no
137 #      AllowTcpForwarding no
138 #      PermitTTY no
139 #      ForceCommand cvs server
root@localhost:~#
```

为了更好的解决行数比较多的文本文档查看问题，我们可以使用more命令进行分屏展示，当使用more命令查看的文本文件，我们可以分为几个部分查看，每个部分按空格键或者回车就可完成切换查看。

```
more /etc/ssh/sshd_config
// 空格键为换屏查看，enter键为换行查看
```

我们在使用more命令来查看文本文件时，只能向下进行查看，如果跳过了想查看的内容，就得退出重新执行一遍，所以非常麻烦。为了解决这个问题，我们使用less命令来查看文件，它具备more命令的空格键和enter键查看的功能，同时还能用**上下键**移动光标查看对应的内容。但是它不会自动的退出文本内容，而是需要手动执行**q**命令退出。

```
less /etc/ssh/sshd_config
```

2.5.3.head和tail命令-查看文件开头或末尾的部分内容

head和tail命令用于显示文件的局部内容。默认条件下，head显示前10行内容，tail显示后10行内容。

例如：查看/etc/passwd文件的前10行和后10行内容：

```
head /etc/passwd
tail /etc/passwd
```

命令选项：

- **-n 数字：展示指定数字的几行内容**

例如：查看/etc/passwd文件前3行的内容：

```
head -n 3 /etc/passwd
也可以写成：head -3 /etc/passwd
```

- **-f:实时显示文件内容的增量**

在生产环境下，tail命令更多的是查看日志文件，以便观察访问数据，服务器信息等。配合“-f”选择跟踪日志文件末尾的变化，实时显示更新内容。

例如：查看系统公共日志文件/var/log/messages的最后10行，并在末尾跟踪显示该文件中的实时更新内容(Ctrl+C键终止)：

```
tail -f /var/log/messages
```

为了方便大家的理解，我们可以在MobaXterm中打开两个连接的窗口，执行如下命令查看具体变化：

```
// 1.在窗口1中
echo 'Hello' > test.txt           // 向test.txt文档中执行增加内容Hello
// 2.在窗口2中
tail -f test.txt                 // 查看test.txt
// 3.在窗口1中
echo 'world' >> test.txt          // 再次向test.txt中增加内容
// 4.在窗口2中
tail -f test.txt                 // 再次查看test.txt中的变化
```

2.5.4.wc命令-文件内容统计

wc(word count)命令是用来统计指定文件中的行数，单词数，字节数的。

```
wc /etc/resolv.conf
// 3 8 73 /etc/resolv.conf --3行，8个单词，73个字节
```

wc命令的常用选项：

- **-l:统计行数；-w:统计单词数；-c:统计字节数。**


```
wc -l /etc/resolv.conf
wc -w /etc/resolv.conf
wc -c /etc/resolv.conf
```

2.5.5.echo命令-输出指定内容

echo命令常用来输出指定的字符串或者变量的值。

例如：在屏幕上输出Hello World:

```
echo "Hello world"
// 我们常配合>, >> 重定向符号使用
echo "Hello world" > a.txt // 先写进a.txt
cat a.txt                 // 再查看内容
```

例如：新建一个变量day，赋值为Monday，再输出变量day的内容

```
day=Monday
echo $day
// 通常在变量前面加上**$符号**，可以引用一个变量的内容。
a=500
b=100
echo [$a+$b]           // 600
// 输出环境变量SHELL
echo SHELL
echo $LANG
```

```
[root@localhost ~]# echo Hello World
Hello World
[root@localhost ~]# a=500
[root@localhost ~]# b=100
[root@localhost ~]# echo [$a+$b]
600
[root@localhost ~]# day=Monday
[root@localhost ~]# echo $day
Monday
[root@localhost ~]#
```

2.5.6.grep命令-文件内容查找

grep命令用于在文本文件中查找并显示包含指定字符串的所在行。通过该命令。我们可以从杂乱的信息中找到所需要的部分。

语法：

```
grep [选项] 查找条件 目标文件
```

例如：在/etc/passwd中查找包含“root”字符串的行，grep会将匹配到的字符串标记为红色。

```
grep 'root' /etc/passwd
// root:x:0:0:root:/root:/bin/bash
// operator:x:11:0:operator:/root:/sbin/nologin
```

注意:grep命令不支持"*"和"?"这些普通意义的通配符，而是通过正则表达式来设置所要查找的条件。正则表达式定义了很多不同意义的符号，如符号"^"表示以什么字符开头，符号"\$"表示以什么字符结尾。如"^word"表示以word开头，"word\$"表示以word结尾。

需要说明的是，如果grep所使用的查找关键字中不包含正则表达式或是空格等特殊符号，那么关键字是否加引号都无所谓，如果关键字出现了这些符号，建议给这些关键字加上引号。

例如：在/etc/passwd查找以“root”开头的行：

```
grep "^root" /etc/passwd
// root:x:0:0:root:/root:/bin/bash
```

grep命令的常用选项：

- **-n:**输出符合查找条件的行及其行号：

```
grep -n 'root' /etc/passwd
// 1:root:x:0:0:root:/root:/bin/bash
// 10:operator:x:11:0:operator:/root:/sbin/nologin
```

- **-v:**反转查找，输出与查找条件不符合的行：

```
grep -v "^#" /etc/passwd // 查找所有不是以#开头的行
-----
grep -v "$" /etc/ssh/sshd/sshd_config // 查找不是空白行的内容
```

- **-i:**不区分大小写：i=ignore

```
grep -i 'a' test.txt // 在文件中查找不区分大写小写有a的行
```

- **-w:**精确匹配单词：

```
echo "The num is 10" > test2.txt // 写内容进test2.txt
cat test2.txt
echo "The number is 10" >> test2.txt // 追加
grep -w 'num' test2.txt
```

- **-r:**递归查找：

通过-r选项可以指定目录及其子目录中查找指定的关键字，有时我们想找一些字符串，但又不知道它在哪个文件中，就可以通过这个选项来实现。

```
grep -nr "DNS" /etc/ssh // 找出包含DNS的所有文件及行号
// /etc/ssh/ssh_config:30:# GSSAPITrustDNS no
// /etc/ssh/sshd_config:115:#UseDNS yes
```

2.5.7.diff命令-文件内容对比

diff命令用于对比多个文件之间的差异，这是系统安全防范中非常重要的。比如黑客入侵系统后，往往会修改一些配置，从而留下一些后门。所谓系统运维人员最好将重要文件备份，然后定期执行diff命令进行对比，从而发现是否被改动过。

```
cp .bashrc .bashrc.bak          // 备份
ls -a                            // 查看包括隐藏文件的所有文件
echo "cd/tmp" >> .bashrc         // 写内容进去
diff .bashrc .bashrc.bak        // 对比
// 13d12
// < cd/tmp
```

显示的结果中"a","d","c"分别表示添加、删除、修改的操作。其中以"<"开始的行属于文件1，以">"开始的行属于文件2。