Linux文件管理

# 目录和文件管理

## Linux系统的目录结构

|  |
| --- |
| /usr/bin、/bin：存放所有用户可以执行的命令  /usr/sbin、/sbin：存放只有root可以执行的命令  /root：引导系统的必备文件，文件系统的装载信息以及系统修复工具和备份工具。  /home：用户缺省宿主目录  /proc：虚拟文件系统，存放当前进程信息  /dev：存放设备文件  /lib：存放系统程序运行所需的共享库  /lost+found：存放一些系统出错的检查结果  /tmp：存放临时文件  /etc：系统配置文件  /var：包含经常发生变动的文件，如邮件、日志文件、计划任务等  /usr：存放所有命令、库、手册页等  /boot：内核文件及自举程序文件保存位置  /mnt：临时文件系统的安装点  /media：系统用来挂载光驱等临时文件系统的挂载点 |

## Linux系统的文件类型

文件类型的种类有：

|  |
| --- |
| -：普通文件 d：目录文件 l：符号链接文件  c：字符设备文件 b：块设备文件 s：socket文件  p：管道文件 |
| 普通文件：文本文件，数据文件，可执行二进制文件（通过file 文件名1 …来查看）  目录文件：用于管理和组织系统中大量文件，其存储一组相关文件的位置，大小等。  符号链接文件：指向同一索引节点的那些目录条目。  设备文件：可分为块设备文件和字符设备文件。Linux把每一个I/O设备都看成一个文件。 |

--file命令查看文件类型，file命令用来探测给定文件的类型。

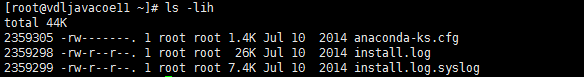
语法：file 选项 参数

选项：

* -b：列出辨识结果时，不显示文件名称；
* -c：详细显示指令执行过程，便于排错或分析程序执行的情形；
* -f<名称文件>：指定名称文件，其内容有一个或多个文件名称时，让file依序辨识这些文件，格式为每列一个文件名称；
* -L：直接显示符号连接所指向的文件类别；
* -v：显示版本信息；



注意：详细信息可以使用ls –lih查看文件的信息主要包括：文件i节点，文件类型-文件拥有者权限-同组用户权限-其他用户权限，文件的链接数，文件所属用户，文件所属组，文件的大小，最后访问的时间



## 文件命名规则

|  |
| --- |
| 1. 文件名应该尽量简单，文件名没有必要超过14个字符 2. 除了/和空格，文件名可以包含任意的ASCII字符。 3. 习惯上应该使用\_或者.来区别文件的类型。不应该有特殊字符（shell有特殊含义） 4. 文件名应该避免使用空格、制表符或者其他控制字符。 5. 为了好的管理，相同文件文件应该使用同样的后缀或者扩展名。 6. 系统区分文件名的大小写 7. 以.开头的文件时隐含文件。默认情况下，shell通配符不匹配这类隐藏文件。 |

## 文件名的通配符

|  |
| --- |
| --作用：用来匹配文件名）  ?：(匹配一个任意字符)  \*：(匹配一个或者多个任意字符)  []：(匹配中括号中任意一个字符)  [-]：(-代表一个范围)  [^]：(逻辑非，不是括号中的任意一个字符) |

# 文件的权限

## 文件和目录的权限

### 一般权限

文件一般权限：

* r：可以查看文件的内容
* w：可以修改文件的内容
* x：可以执行文件

目录的一般权限：

* r：可以列出目录中的内容ls
* w：可以在目录中创建删除文件
* x：可以进入目录cd

8进制表示权限：4（r）,2（w）,1（x）。

### 特殊权限

setUID：执行文件的用户以该文件所属用户的权限去执行。 s 这个标志出现在文件拥有者的 x 权限上。

* SetUID只有可执行的二进制程序才能设定;
* 命令执行者要对该程序拥有x执行权限；
* 命令的执行者在执行该程序的时候获取该程序文件的属主的身份（普通用户暂时获取文件的所有者权限）；

注意：

* 危险的SetUID:关键的目录应该控制写权限，比如/，/usr等；
* 对用户密码准守三规则；
* 对系统中默认具有SetUID权限的文件做一列表，定时检查有没有之外的文件被设置了SetUID权限。

setGID:：对于二进制文件，普通用户执行二进制文件的时候，组身份变成二进制文件所属组；对于目录，如果普通用户在这个目录中新建文件，文件的所属组是目录所属组。

Sticky Bit: 目前只对目录有效；普通用户对目录拥有w和x权限，即普通用户可以进入用户并进行写的权限；如果没有粘着位，用户可以删除此目录的所有文件，包括其他用户建立的文件，一旦赋予了粘着位，除了root可以删除所有权限，普通用户就算有写权限，只能删除自己建立的文件，但是删除其他用户建立的文件。

特殊权限的8进制表达:4(SetUID),2(SetGID),1(t)

### 给文件设置特殊权限

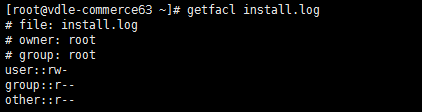
|  |
| --- |
| chmod u+s xxx # 设置setuid权限  chmod g+s xxx # 设置setgid权限  chmod o+t xxx # 设置stick bit权限，针对目录  chmod 4775 xxx # 设置setuid权限  chmod 2775 xxx # 设置setgid权限  chmod 1775 xxx # 设置stick bit权限，针对目录  查找SetUID程序： find / -perm -4000 -o -perm -2000 |

注意：有时你设置了s或t  权限，你会发现它变成了S或T，这是因为在那个位置上你没有给它x(可执行)的权限，这样的话这样的设置是不会有效的，你可以先给它赋上x的权限，然后再给s或t  的权限。

## ACL访问控制权限

### getfacl命令

getfacl 文件名



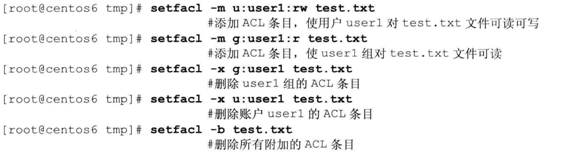
### setfacl命令

作用：设置文档访问控制列表

用法：setfacl [选项] [{-m|-x} acl条目] 文件或者目录

选项：

* -b：删除所有的附带ACL条目
* -k：删除默认的ACL条目
* -m：添加ACL条目
* -x：输出指定的acl条目
* -R：递归处理所有的子文件与子目录。
* m：是用来指定最大有效权限的。如果我给用户赋予了ACL权限，是需要和mask权限做”相与”才能得到用户的真正权限（不影响所有者权限）。 例如：setfacl -m m:rx 指定文件或者目录
* d：如果给父目录设定了默认的acl权限，那么父目录中所有新建的子文件都会继承父目录的acl权限。 setfacl -m d:u:用户名:权限 文件名



## 修改文件或目录的权限

### chmod命令

|  |
| --- |
| 命令名称：chmod  功能描述：改变文件或目录权限  语法：chmod [{augo}{+-=}{rwx}] [文件或目录]  [mode=421 ] [文件或目录] |
| -- 范例  #赋予文件file1所属组写权限  $ chmod g+w file1  #设定目录dir1为所有用户具有全部权限  $ chmod 777 dir1 |

### umask命令

|  |
| --- |
| 命令名称：umask  功能描述：显示、设置文件的缺省权限  语法：umask [-S] maskcode   * -S 以rwx形式显示新建文件或目录缺省权限   范例：$ umask  #文件最大权限666，目录最大权限777。用文件和目录的最大权限减去#umask设置的缺省权限，得到的就是目录和文件的默认创建时有的权限。  $ umask –S |

### chown命令

|  |
| --- |
| 命令名称：chown  功能描述：改变文件或目录的所有者  语法：chown –R [用户] [文件或目录]  范例：$ chown mysql.mysql file1 |

### chgrp命令

|  |
| --- |
| 命令名称：chgrp  语法：chgrp [用户组] [文件或目录]  功能描述：改变文件或目录的所属组  #改变文件file1的所属组为adm  chgrp adm file1/chgrp –R 用户组 目录 |

# 常用文件相关命令

## 文件显示命令

### cat 命令

|  |
| --- |
| 命令名称：cat  语法：cat [-n] [文件名]  功能描述：显示文件内容  范例：$ cat /etc/issue  $ cat /etc/services(-n可以对行号进行编号) |

### more和less命令

|  |
| --- |
| 命令名称：more  功能描述：分页显示文件内容  语法：more [文件名]   * (空格) 或f 显示下一页 * (Enter) 显示下一行 * q或Q 退出   范例： $ more /etc/services |

### head命令

|  |
| --- |
| 指令名称：head  功能描述：查看文件的前几行,默认前10行  语法：head -num [文件名]  -num 显示文件的前num行  范例：$ head -20 /etc/services |

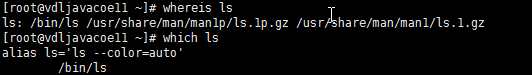
### tail命令

|  |
| --- |
| 指令名称：tail  功能描述：查看文件的后几行  语法：tail -num [文件名]   * -num 显示文件的后num行 * -f 动态显示文件内容   范例： tail –f 文件名 |

## 文件搜索命令

### which命令

|  |
| --- |
| 命令名称：which  功能描述：显示系统命令所在目录  语法：which [命令名称] （同whereis）  范例：$ which ls |



### find命令

|  |
| --- |
| 命令名称：find  功能描述：查找文件或目录  语法：find [搜索路径] [搜寻关键字]  常用参数：   * -name: 查找时文件名大小写敏感。 * -iname: 查找时文件名大小写不敏感。 * -perm mode: 文件权限正好符合mode(mode为文件权限的八进制表示)。 * -perm +mode: 文件权限部分符合mode。如命令参数为644(-rw-r--r--)，那么只要文件权限属性中有任何权限和644重叠，这样的文件均可以被选出。 * -perm -mode: 文件权限完全符合mode。如命令参数为644(-rw-r--r--)，当644中指定的权限已经被当前文件完全拥有，同时该文件还拥有额外的权限属性，这样的文件可被选出。 * -user：按照文件属主来查找文件（-nouser） * -group：按照文件所属组来查找文件（-nogroup） * -size [+/-]100[c/k/M/G]: 表示文件的长度为等于[大于/小于]100块[字节/k/M/G]的文件。 * -empty: 查找空文件。 * -type：按照文件的类型来查找（f,d,b,c,p,l,s） * -mtime +n-n：-n代表n天以内，+n代表n天以前。 * -ctime -n[+n]: 找出文件更改时间在n日之内[之外]的文件。 * -mtime -n[+n]: 找出修改数据时间在n日之内[之外]的文件。 * -amin -n[+n]: 找出文件访问时间在n分钟之内[之外]的文件。 * -cmin -n[+n]: 找出文件更改时间在n分钟之内[之外]的文件。 * -mmin -n[+n]: 找出修改数据时间在n分钟之内[之外]的文件。 * -print： find命令将匹配的文件输出到标准输出；   逻辑：-o(or),-a(and,缺省),！(表示非)  操作：-exec/ok 命令 {} \;  -exec/-ok的区别：-ok会逐个询问 |
| #在目录/etc中查找文件init  $ find /etc -name init  #在根目录下查找大于100MB的文件  $ find / -type f -size +200M  #在根目录下查找所有者为sam的文件  $ find / -user sam  #在/etc下查找24小时内被修改过属性的文件和目录  $ find /etc -ctime -1  #在/etc下查找inittab文件并显示其详细信息  $ find /etc -name inittab -exec ls -l {} \; |
| xargs命令的使用  在使用find命令的-exec选项处理匹配到的文件时， find命令将所有匹配到的文件一起传递给exec执行。但有些系统对能够传递给exec的命令长度有限制，这样在find命令运行几分钟之后，就会出现溢出错误。错误信息通常是“参数列太长”或“参数列溢出”。这就是xargs命令的用处所在，特别是与find命令一起使用。 find命令把匹配到的文件传递给xargs命令，而xargs命令每次只获取一部分文件而不是全部，不像-exec选项那样。这样它可以先处理最先获取的一部分文件，然后是下一批，并如此继续下去。  find . -name 'hell\*' | sort | xargs wc –l |

### locate命令

|  |
| --- |
| 指令名称：locate  功能描述：寻找文件或目录  语法：locate [搜索关键字] |
| updatedb(不包含临时文件)  locate -i（不区分大小写） filename |

## 操作文件目录命令

### touch命令

|  |
| --- |
| 命令名称：touch  语法：touch [文件名]  功能描述：创建空文件  范例：$ touch newfile |

### mkdir命令

|  |
| --- |
| 命令名称：mkdir  功能描述：创建新目录  语法：mkdir [目录名]  #使用-p可以建立多级目录 ,-m 700设置目录的存取权限  范例：$ mkdir –p newdir |

### cp命令

|  |
| --- |
| 命令名称：cp  功能描述：复制文件或目录  语法：cp –Rpu [源文件或目录] [目的目录]   * -R 复制目录 * -p 保持备份目录及文件属性 * -u增量备份 |
| #将文件file1、file2复制到目录dir1  $ cp file1 file2 dir1  #将dir1下的所有文件及子目录复制到dir2  $ cp -R dir1 dir2 |

### scp命令

|  |
| --- |
| 命令名称：scp  语法：scp(选项)(参数)  选项：   * -q：不显示复制进度； * -r：以递归方式复制。   参数：   * 源文件：指定要复制的源文件。 * 目标文件：目标文件。格式为user@host：filename（文件名为目标文件的名称）。 |
| --从远处复制到本地  scp -r root@10.10.10.10:/opt/soft/mongodb /opt/soft/  --上传本地文件到远程机器指定目录  scp /opt/soft/nginx-0.5.38.tar.gz root@10.10.10.10:/opt/soft/scptest |

### mv命令(剪切)

|  |
| --- |
| 命令名称：mv  功能描述：移动文件、更名  语法：mv [源文件或目录] [目的目录] |
| #将当前目录下文件file1更名为file3  mv file1 file3  #将文件file2移动到目录dir2下  mv file2 dir2 |

### rm命令

|  |
| --- |
| 命令名称：rm  功能描述：删除文件  语法：rm -rf [文件或目录]   * -r 删除目录 * –f无需逐一确认 * –i要逐一访问   使用： rm -rf dir1/file |

### rmdir 命令

|  |
| --- |
| 命令名称：rmdir  功能描述：删除目录  语法：rmdir -p 目录  -p 当子目录被删除后它也成为了空目录，则一起删除  范例：rmdir bin/ |

## 改变和显示工作目录

### ls命令

|  |
| --- |
| 命令名称：ls  功能描述：显示目录文件  语法：ls 选项[-ald] [文件或目录]   * -a 显示所有文件，包括隐藏文件 * -l 详细信息显示 * -d 查看目录属性   常用：ls –lih 文件或目录路径 |

### cd 命令

|  |
| --- |
| 命令名称：cd  功能描述：切换目录  语法：cd [目录]  范例：$ cd / 切换到根目录  $ cd .. 回到上一级目录 |

### pwd命令

|  |
| --- |
| 命令名称：pwd  功能描述：显示当前所在的工作目录  语法：pwd  范例：$ pwd  /etc/rc5.d |

## 链接文件命令

### ln命令

命令名称：ln

功能描述：产生链接文件

语法：ln -s [源文件] [目标文件]

* -s 创建软链接，默认是硬链接

特点：

* 软连接的权限：lrwxrwxrwx
* 软连接的箭头指向源文件。
* 硬链接相当于cp -p+同步
* 硬链接共享同一个inode(多个文件可以共享一个inode）
* 硬链接可以通过inode来识别
* 硬链接不能跨分区
* 硬链接不能针对目录使用

注意：符号链接相当于window系统中的快捷方式。其内容只是一个实际存在文件的路径描述。符号链接文件与目标文件是两个独立的文件，有着各自的i节点和数据块。硬链接则是将两个或者多个文件指向同一个i节点。硬链接具有不同文件名和相同的i节点。

## 比较文件内容的命令

### comm命令

|  |
| --- |
| 功能：comm.命令对两个已经排好序的文件进行逐行比较。  格式：comm [-123] 文件1 文件2   * 选项-123的含义是不显示comm输出中的第一列、第二列和第三列。   #第一列表示只有文件1中才有的行，第二列表时只有文件2才有的行。  #第三列表示两个文件中都有的行。  例子：comm 1.txt 2.txt |

### diff命令

|  |
| --- |
| 功能：diff命令逐行比较两个，列出不同之处  格式：diff 选项 文件1 文件2  选项：  -b忽略行尾的空格，并且字符串的空白符都被看成是相等的  -c输出格式是带上下文的三行格式  -r当文件1,2都是目录的时候，递归比较各子目录  例子：diff 1.txt 2.txt |

## 文件的压缩和打包

打包：将许多文件的目录变成一个总的文件，该文件的体积并不会缩小。

压缩：是将一个大的文件通过一些压缩算法变成一个小的文件。

Linux系统中很多压缩程序只能针对一个文件进行压缩，对于文件夹就只能先打包成一个大的文件，然后进行压缩。Linux下常用的压缩工具有bzip2/gzip/zip，对应的解压工具bunzip2/gunzip/unzip。

### 压缩命令：gzip

|  |
| --- |
| 命令名称：gzip  功能描述：压缩文件，压缩后文件格式：.gz  语法：gzip 选项[文件] |

### 解压命令：gunzip

|  |
| --- |
| 命令名称：gunzip  语法：gunzip 选项[压缩文件]  功能描述：解压缩.gz的压缩文件  范例：$ gunzip file1.gz |

### 压缩命令bzip2

|  |
| --- |
| 命令名称：bzip2  语法：bzip2 选项[-k] [文件]   * -k 产生压缩文件后保留原文件   功能描述：压缩文件  压缩后文件格式：.bz2  范例：$ bzip2 -k file1 |

### 解压命令bunzip2

|  |
| --- |
| 命令名称：bunzip2  功能描述：解压缩  语法：bunzip2 [-k] [压缩文件]   * -k 解压缩后保留原文件   范例：$ bunzip2 -k file1.bz2 |

### 压缩命令zip

|  |
| --- |
| 命令名称：zip  语法：zip 选项[-r] [压缩后文件名称] [文件或目录]  功能描述：压缩文件或目录   * -r :压缩目录 * -q:不显示过程   压缩后文件格式：.zip  例如： /home目录下面的mydata目录压缩为mydata.zip  zip -r mydata.zip mydata #压缩mydata目录 |

### 解压命令unzip

|  |
| --- |
| 命令名称：unzip  功能描述：解压.zip的压缩文件  语法：unzip -nq [压缩文件]   * -n：覆盖已经存在的文件 * -q：安静模式   范例：$ unzip test.zip  例如：把/home目录下面的mydata.zip解压到mydatabak目录里面  unzip mydata.zip -d mydatabak |

### tar的使用

命令名称：tar

功能描述：打包目录

语法：tar 选项[zcxvf] [目录]

* -z 打包同时压缩
* -c 产生.tar打包文件
* -x 解包.tar文件
* -v 显示详细信息
* -f 指定压缩后的文件名
* -C指定解压到哪个目录

|  |
| --- |
| #创建压缩的tar文件  tar –zcvf filename.tar 目录或者文件  #显示tar文件的内容  tar -ztvf host.tar.gz  # tar包中还原  tar -zxvf host.tar.gz -C .  #向tar包中追加文件  tar rvf filename.tar 文件或者目录 |

### cpio命令

|  |
| --- |
| --备份  find /etc/ -name \*.conf | cpio -cov > /tmp/conf.cpio  --还原  cpio --absolute-filenames -icvu < /tmp/conf.cpio |

# 帮助命令

## 帮助命令man

|  |
| --- |
| 命令名称：man  功能描述：获得帮助信息  语法：man [命令或配置文件]  范例：$ man ls |

## 帮助命令info

|  |
| --- |
| 指令名称：info  功能描述：获得帮助信息  语法：info [任何关键字]  范例：$ info ls  查看ls指令的帮助信息 |

## 帮助命令whatis

|  |
| --- |
| 指令名称：whatis apropos makewhatis  语法：whatis apropos [任何关键字]  功能描述：获得索引的简短说明信息  范例：$ whatis ls  $ apropos fstab 相当于man –k  范例：# makewhatis |

## 帮助命令的比较

“--help”选项并不是一个“独立”的工具。作为一种命令的选项，它可以用来修改工具或者命令的工作方式。命令的选项通常由一个或两个连字符后跟一个或多个字母来指定。选项出现在所调用的工具名后，用空格隔开。工具的其它参数都跟在选项后，也用空格隔开。“--help”选项就像分页程序“| less”一样，它所提供的是一种快捷、高效的帮助。

Man和info就像两个集合，它们有一个交集部分，但与man相比，info工具可显示更完整的最新的GNU工具信息。若man页包含的某个工具的概要信息在info中也有介绍，那么man页中会有“请参考info页更详细内容”的字样。通常情况下，man工具显示的非GNU工具的信息是唯一的，而info工具显示的非GNU工具的信息是man页内容的副本补充。