

Linux常用命令

文件处理命令

ls命令

说明	内容	备注
命令名称	ls	
英文原意	list	
命令所在路径	/bin/ls	
执行权限	所有用户	
功能描述	显示目录文件	
语法	ls [OPTION]...[FILE or DIR]	

参数简写	参数全称	参数作用	备注
-a	--all	显示所有文件，包括隐藏文件	
-l	无	显示详细信息	l 的意思就是 long
-d	--directory	只显示目录信息，而不查看目录的内容	
-h	--human-readable	人性化显示文件大小(k/m/g)	
-i	--inode	查看文件的 index number	每一个文件的 inode号是唯一的
-t	无	按照最新的修改时间进行排序	t 可以理解为 time

eg: ls -alh 显示当前目录下的所有文件（包括隐藏）的详细信息。

ls 文件信息说明

eg：-rw-r--r-- 1 chonepieceyb chonepieceyb 2.9K 3月 27 22:24 sources.list

文件详细内容依次为

- **文件类型**：- 二进制文件、 d 目录 、 l 软连接文件
- **权限**：r 读 w 写 x 执行，以 rwx为一组重复三次，从左到右分别表示 u(所有者)g(组)o(其他人 other) 的 读写执行权限，字母出现了表示有权限，字母没出现 (-) 表示没权限
- **引用技术** :在硬链接使用？
- **所有者**
- **所属组**
- **文件大小** 默认以字节为单位，可以加上 -h 参数采用人性化显示
- **最新修改时间**
- **文件名**

目录处理命令

mkdir 命令

说明	内容	备注
命令名称	mkdir	
英文原意	m ake d irectory	
命令所在路径	/bin/mkdir	
执行权限	所有用户	
功能描述	显示目录文件	
语法	mkdir [option]... [DIRECTORY1] [DIRECTORY2]...	后面的 dir1 dir2 表示可以同时创建多个目录，中间用空格隔开

参数简写	参数全称	参数作用	备注
-p	--parents	递归创建，也就是如果当被创建的目录不存在的时候描，会自动创建其父目录	

eg: mkdir -p /home/temp/file1 /home/temp/file2

cd 命令

说明	内容	备注
命令名称	cd	
英文原意	c hange d irectory	
命令所在路径	shell内置命令	
执行权限	所有用户	
功能描述	切换目录	
语法	cd [-L -P] [directory]	-L

ps：

- 表示当前目录 eg cd ./temp (打开当前目录的 temp).
- .. 表示上一级目录 eg cd .. (返回上一级目录)

pwd 命令

说明	内容	备注
命令名称	pdw	
英文原意	p rint w orking d irectory	
命令所在路径	/bin/pwd	
执行权限	所有用户	
功能描述	显示当前目录的绝对路径	
语法	pwd [OPTION]	参数可以查看man手册

rmdir

说明	内容	备注
----	----	----

说明	内容	备注
命令名称	rmdir	
英文原意	remove empty directories	
命令所在路径	/bin/rmdir	
执行权限	所有用户	
功能描述	删除指定的空目录	只能删除空目录，如果目录下存在内容，这个命令就用不了
语法	rmdir [OPTION] DIRECTORY	参数可以查看man手册

cp

说明	内容	备注
命令名称	cp	
英文原意	copy	
命令所在路径	/bin/cp	
执行权限	所有用户	
功能描述	复制文件或者目录	该命令可以实现在复制的同时重命名，如果dst是一个dir的话就只赋值，如果dst是文件名就会将原文件重命名
语法	cp [OPTION]... SOURCE... DIRECTORY	... 表示有多个，这里表示可以把多个文件复制到一个文件夹下。 语法格式不只一种详情查man

参数简写	参数全称	参数作用	备注
-r\ -R	--recursive	递归复制， 加上这个参数可以复制一个文件夹的所有内容	
-p	--preserve[=ATTR_LIST]	保留文件的属性	-p 等同与 --preserve=mode,ownership,timestamps 保留所有信息
-f	--force	如果一个文件存在，把它删除， 继续尝试复制操作	
-n	--no-clobber	不覆盖已经存在的文件	这一个参数会把在其之前出现的 -i 参数覆盖
-i	--interactive	覆盖之前提示询问是否覆盖	这一参数会覆盖其之前出现的 -n 参数

mv

说明	内容	备注
命令名称	mv	
英文原意	move	
命令所在路径	/bin/mv	
执行权限	所有用户	

说明	内容	备注
功能描述	剪切文件或者目录	mv 不需要 -r参数也可以剪切文件夹，可以在剪切的同时改名。 在同一个目录下进行mv操作等价于重命名
语法	mv [OPTION]... SOURCE... DIRECTORY	... 表示有多个，这里表示可以把多个文件剪切到一个文件夹下。 语法格式不只一种详情查man

参数简写	参数全称	参数作用	备注
-f	--force	覆盖的时候不提示	如果同时指定多个 -f -n -i 参数只有最后一个生效
-n	--no-clobber	不覆盖已经存在的文件	
-i	--interactive	覆盖之前提示询问是否覆盖	

rm

说明	内容	备注
命令名称	rm	
英文原意	remove	
命令所在路径	/bin/rm	
执行权限	所有用户	
功能描述	删除文件	删除目录需要加上 -r Linux下没有回收站，不要乱用rm， 误删除的话不要做过多读写操作（还是有希望可以从硬盘恢复数据， 硬盘读写操作越多，希望越小）
语法	rm [OPTION]... [FILE]...	... 表示有多个

参数简写	参数全称	参数作用	备注
-r\ -R	--recursive	递归删除，加上这个参数可以删除一个文件夹的所有内容	
-f	--force	强制删除，忽略不存在的文件，删除不提示	
-n	--no-clobber	不覆盖已经存在的文件	这一个参数会把在其之前出现的 -i 参数覆盖
-i	无	每次删除的时候都询问是否删除	
-I	无	当一次删除超过3个文件的时候，或者发生递归删除 （删除文件夹）的时候询问是否删除	

文件处理命令

touch 命令

说明	内容	备注
命令名称	touch	
英文原意	torch - change file timestamps	
命令所在路径	/bin/touch	

说明	内容	备注
执行权限	所有用户	
功能描述	创建新文件/将每个文件的access and modification time 更新为当前的时间	如果 FILE 不存在就会新建文件
语法	touch [OPTION]... [FILE]...	

参数简写	参数全称	参数作用	备注
-c	--no-create	不创建任何文件	
-a	无	只改变 access time	
-m	无	只改变 modificatioin time	

cat

说明	内容	备注
命令名称	cat	
英文原意	concatenate files and print on the standard output	
命令所在路径	/bin/cat	
执行权限	所有用户	
功能描述	显示文件内容(tac 命令可以倒着看文件)	其实 cat 命令 可以和 vim 的 r 结合 把一个文件的内容导入到 vim
语法	cat [OPTION]... [FILE]...	...表示可以同时看多个文件

参数简写	参数全称	参数作用	备注
-n	--number	显示行号	
-b	--number-nonblank	number nonempty output lines, overrides -n	

more

说明	内容	备注
命令名称	more	
命令所在路径	/bin/more	
执行权限	所有用户	
功能描述	分页查看文件内容	
语法	more [OPTION] [FILE]...	

操作	操作效果	备注
SPACE\f	翻页	翻到下一页
b	翻回上一页	
RETURN	换到下一行	回车

操作	操作效果	备注
q/Q	退出	

less

说明	内容	备注
命令名称	less	
命令所在路径	/bin/less	
执行权限	所有用户	
功能描述	分页查看文件内容 (和more类似但功能更多)	这里只是简单地列举了一些用法，具体的请查阅man手册， man手册好像使用less进行查看
语法	less [FILE]...	...表示可以同时查看多个文件

操作	操作效果	备注
SPACE\f	翻页	翻到下一页
b	翻回上一页	
RETURN	换到下一行	回车
/pattern	查找正则表达式pattern的内容	
n	查找下一个	
N	查找上一个	
:n	查看下一个文件	如果在命令行同时打开多个文件的话
:p	查看上一个文件	
e:[file]	查看一个新的文件（由file指定）	
q/Q	退出	

ps: less 可以用 上下左右方向键进行查看，也可以用 page up page down 进行上下翻页

head

说明	内容	备注
命令名称	head	
命令所在路径	/usr/bin/head	
执行权限	所有用户	
功能描述	显示文件的头几行	
语法	head [OPTION]... [FILE]...	...表示可以同时查看多个文件

参数简写	参数全称	参数作用	备注
-n 行数num	--lines 行数num	显示前面的 num 行，如果不指定的话默认显示 10 行	

参数简写	参数全称	参数作用	备注
-n +num	--lines +num	从头开始显示除了最后num行的内容（也就是从倒数第num行开始往前显示）， 如果这里用 -num 表示倒数 num行	

tail

说明	内容	备注
命令名称	tail	
命令所在路径	/usr/bin/tail	
执行权限	所有用户	
功能描述	显示文件的末尾几行	
语法	tail [OPTION]... [FILE]...	...表示可以同时查看多个文件

参数简写	参数全称	参数作用	备注
-n 行数num	--lines 行数num	显示末尾的 num 行，如果不指定的话默认显示 10 行	
-n +num	--lines +num	现实从第num行开始 到文件结尾的内容 如果这里用 -num 表示倒数 num行	
-f	--follow	可以动态现实（可以很方便的动态显示日志文件）	

tail、head、cat 组合应用

这里可以将 cat tail head 命令组合起来实现 查看文件指定行数范围的内容
eg: cat file |tail -n +10 | head -n 10 查看 file 从 第10行 到第 19行的内容
cat file |head -n 3000| tail -n +1000 查看 file 1000 - 3000 行的内容

ps: 将上述命令 和 vim 的 :r (导入文件) 以及 : !cmd (不退出vim执行命令行命令)结合起来 可以轻松实现将指定文件的指定范围的内容 导入 vim

ln

说明	内容	备注
命令名称	ln	
英文原意	link	
命令所在路径	/bin/ln	
执行权限	所有用户	
功能描述	生成链接文件	
语法	ln [OPTION]...[-T] TARGET LINK_NAME / ln [OPTION]... TARGET / ln [OPTION]... TARGET...DIRECTORY	这个命令有很多种模式(详情请查阅man), 第一种表示生成TARGET的链接, 链接名是LINK_NAME, 第二种表示在当前目录下生成一个TARGET的链接, 第三种表示在 DIRECTORY 目录下为多个TARGET 生成链接。

参数简写	参数全称	参数作用	备注
------	------	------	----

参数简写	参数全称	参数作用	备注
-f	--force	如果目标文件存在就删除它(和mv，cp的类似)	
-i	--interactive	要删除目标文件的时候，出现提示	
-s	--symbolic	生成软连接（符号链接）	
-n	--no-dereference	treat LINK_NAME as a normal file if it is a symbolic link to a directory，（这块不是很懂）	

软连接和硬链接

软连接

软连接又成为符号链接（symbolic link),在Linux中软链接等价于windows中的快捷方式。软链接例子：

```
eg: lrwxrwxrwx 1 chonepieceyb chonepieceyb 13 4月 3 22:55 link.soft -> file_tool.cpp
```

- 1 表明该文件的类型是软连接
- 这里可以看到软连接的 u g o 的权限都是 rwx，这个可以理解为，软链接只是一个快捷方式，真正的权限是由源文件决定的
- 软连接一般都比较小，这里只有 3个字节
- 可以看到最后的一项表明了软连接的目标
- 硬链接没有专门的类型
- 删除target，软链接无法定位到源文件，会报错

硬链接

硬链接可以看成是**target**的一个**copy**但是不完全相同，硬链接的文件属性和target完全相同，硬链接和target是绑定在一起的（换言之target 变化硬链接的内容也会发生变化）硬链接可以同步更新，删除源文件不影响硬链接，删除硬链接也不影响源文件。

硬链接实例；

```
源文件：-rw-r--r-- 2 chonepieceyb chonepieceyb 3358 4月 3 23:07 file_tool2.cpp
```

```
源文件 ionde: 533132 file_tool2.cpp
```

```
硬链接：-rw-r--r-- 2 chonepieceyb chonepieceyb 3358 4月 3 23:07 file_tool2.hard
```

```
硬链接 ionde：533132 file_tool2.hard
```

- 硬链接和其源文件的信息完全相同
- 硬链接和源文件可以同步更新
- 硬链接和源文件可以相互独立（删除互相不影响）
- 硬链接和源文件对应的inode是相同的
- 在文件详细信息的 索引次数（这里是 r--后的 2）表明 该 inode节点的硬链接的个数（在这个例子中有两个，一个是tgt,一个是生成的hard link）
- 只能有一个**inode**但是 一个**inode**可以对应多个文件。

硬链接个人理解：

在linux中，每一个文件有唯一的一个**inode**,但是**inode**可以对应多个文件，在生成硬链接之后，硬链接和其原本的target，都对应着同一个 **inode**这也就是为什么他们可以同步的原因，删除一个硬链接并不影响其它的硬链接。一个target生成硬链接之后，个人认为这个硬链接和target本质上是等价并且互相独立的，他们都可以看成是同一个**inode**节点的**硬链接**

硬链接可以用来备份重要文件！

ps: 硬链接不能跨分区(用windows来打比方不能同时放在C盘和D盘)，硬链接不能作用在目录（超级用户可以通过某些参数实现）

权限管理命令

chmod

说明	内容	备注
----	----	----

说明	内容	备注
命令名称	chmod	
英文原意	change the permissions mode of a file	
命令所在路径	/bin/chmod	
执行权限	所有用户	只有 root 用户和文件的所有者可以改变文件的权限
功能描述	改变文件或者目录的权限	
语法	<pre>chmod [OPTION]... [{ugoa}{+ -=}{rwx}] FILE... chmod [mode]... FILE...</pre>	<p>这个命令详细的格式可以查看man手册，在这里列出来的， 第一种：u g o 表示 所有者，所属组，其他人。a 表示所有用户。 +-= 分别表示 增加权限，减少权限，把权限设置为。 r、w、x 分别表示读、写、执行（其实不只有rwx,还有其他的选项比如 t X s(查man)） 第二种采用数字表示法，mode的值是一个数字其中:r ---4 (100),w---2 (010),x ---1 (001),rwx可以叠加（用二进制比较好理解，每一位(总共三位)表示 r、w、x权限的一种(可以取0 或者 1)，所以如果权限是 rw-，对应的数字就是 6(110)</p>

参数简写	参数全称	参数作用	备注
-R	--recursive	递归修改权限， 如果在修改文件夹权限的时候加了 -R 那么该文件夹下所有文件的权限也都会被修改	

eg: chmod g= test 把test的 group的权限设置为 ---
chmod o+x test 给test增加 x权限
chmod 755 test 将test的权限设置为 **rwxr-xr-x**

文件权限的深入理解

权限简写	权限	文件	目录
r	读权限	可以查看文件的内容	可以列出目录的内容
w	写权限	可以修改文件的内容	可以在目录中创建，删除文件
x	执行权限	可以执行文件	可以进入目录

ps:要很注意文件的rwx权限和目录的rwx权限的含义不同。
eg: test/test.cpp
test 是一个文件夹，test.cpp是一个文件

假如现在的身份是 u

- 1. test 的 u权限为 rwx
test.cpp的 u权限 为 ---
这时候 仍然**可以**删除 test下的 test.cpp
- 2. test 的 u权限为 r-x,test.cpp的u权限为rwx，这时候u**不能**删除test.cpp

chown

说明	内容	备注
----	----	----

说明	内容	备注
命令名称	chown	
英文原意	change file ownership	
命令所在路径	/bin/chown	
执行权限	所有用户	只有root才能改变一个文件的所有者\所属于=组
功能描述	改变文件或者目录的所有者\或者所属组	
语法	chown [OPTION]...[OWNER][:[GROUP]] FILE...	这个命令我们既可以更改所有者也可以更改所有组，1. 如果只给了 OWNER 那么只有该文件的所有者会被改为 OWNER , 2.如果在 OWNER 后面加了冒号，并且跟上了 GROUP eg chmod root:root file 那么会把所有者改成 OWNER , 所属组改成 GROUP 3.如果只在 OWNER 后面跟上冒号，那么会把所有者改为 OWNER 所属组改为 OWNER的所有组 4 如果只有 冒号+ GROUP eg chmod :root file , 那么只会将所属组改为 GROUP 5 如果只有冒号，那就什么都不改

ps:上述的chown是GUN版本的chown，修改所有者或者所属组的时候，OWNER和GROUP必须存在

参数简写	参数全称	参数作用	备注
-R	--recursive	递归修改	

eg
chmod root:root file 把file 的所有者改为root,所属组也改为root

chgrp

说明	内容	备注
命令名称	chgrp	
英文原意	change file group ownership	
命令所在路径	/bin/chgrp	
执行权限	所有用户	只有root才能改变一个文件的所属组
功能描述	改变文件或者目录的所属组	
语法	chown [OPTION]... GROUP FILE...	这个命令等价于 只加冒号和GROUP 的chown

参数简写	参数全称	参数作用	备注
-R	--recursive	递归修改	

eg: chgrp root file 改变 file的所属组为 root

缺省的所有者和所有组

一个文件创建的时候，所有者默认为创建文件的用户，所属组默认为创建文件用户的缺省组

umask

说明	内容	备注
命令名称	umask	
英文原意	the u ser file-creation mask	
命令所在路径	Shell内置命令`	
执行权限	所有用户	
功能描述	显示，设置文件的缺省权限	非root用户也可以修改umask默认值
语法	umask [-S] [mask]	当不加 [mask]的时候就是显示默认的权限，当加上mask表示设置默认的权限。

参数简写	参数全称	参数作用	备注
-S		使用 rwX 的格式输出	

eg：umask 022 设置默认权限为 rwxr-xr-x

ps：在linux中新建的文件(非目录)**不具有**执行权限，这是出于安全考虑。

umask输出说明

```
chonepieceyb@chonepieceyb-VirtualBox:~/文档/c++learning/c++11/aliasTemplate/testdir$ umask
0022
chonepieceyb@chonepieceyb-VirtualBox:~/文档/c++learning/c++11/aliasTemplate/testdir$ umask -S
u=rwx,g=rX,o=rX
```

这里数字有四位，第一位表示**特殊权限**，后三位分别表示 u g o的默认权限。

数字的含义：

把数字转换成2进制后，有三位，从高位到低为分别表示 **r,w,x**，如果值去1表示**没有**这个权限(这就是mask的含义)，输出的结果相当于一个mask, !mask 与 777 与操作就可以得到相应的权限的编码。

eg：umask 的结果 0 -> 000 表示三种权限都有， 2-> 010 表示没有写权限。