# Linux 基础

Linux 中目录命名规则：

1. 长度不能超过255个字符
2. 不能使用/当文件名，Linux系统是用来当路径分隔符和根来使用的。其他任何字符都可以使用\*来当文件名，也可以使用空格来当文件名。
3. 严格区分大小写。

Linux 能干啥？

1.目录管理 ：列出，切换，创建，显示，删除

ls : 列出文件夹和文件

cd : 切换目录

pwd ：显示文件夹属性

## mkdir

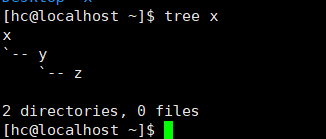
创建空目录

mkdir ./x/y/z 可以写成 mkdir /x/y/z 注意要一级一级的创建

./ 表示相对路径，相对与当前父目录

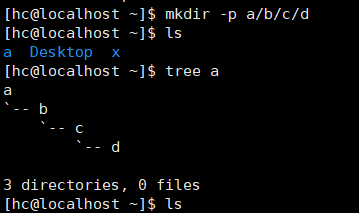
## tree

**查看目录树**

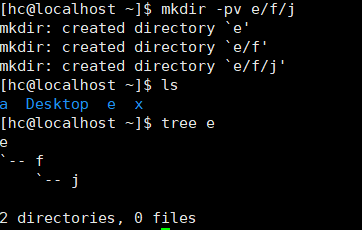


**-p**  （parent 如果没有父级目录，就创建父级目录）

也可以 mkdir –p /x/y/z 就可以一次全部创建成功

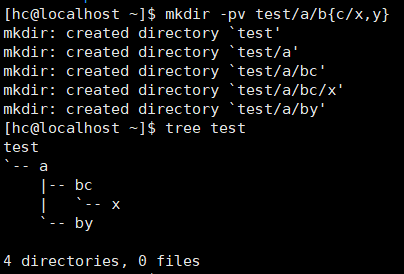


**-v**  (verbose 详细信息,很多命令都带 -v）

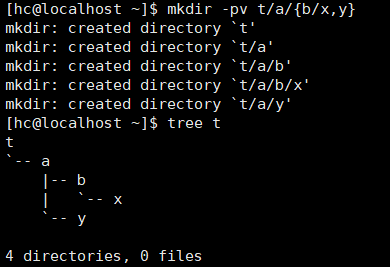


还可以同时创建在一个文件夹下创建两个以上的文件

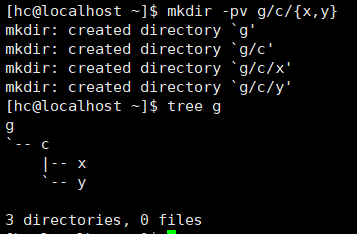
mkdir –pv test/a/b{c/x,y} or mkdir test/a/b{c/x,y} –pv



mkdir –pv t/a/{b/x,y}



mkdir –pv g/c/{x,y}



## { }

命令行展开

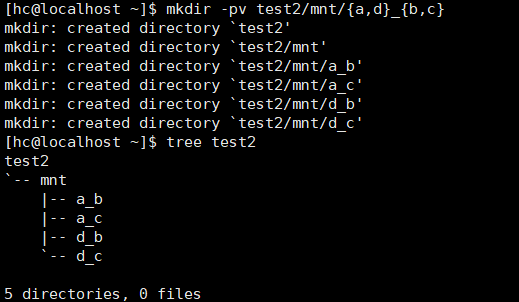
## ~

波浪线展开

创建 a\_b,a\_c,d\_b,d\_c 使用一个命令创建？

（a+d）(b+c) = ab+ac+db+dc

mkdir –pv test2/mnt/{a,d}\_{b,c}



## rmdir

**移除目录 只能移除空目录**

**rm -r**

**rm -rf 都可以删除目录和文件**

## touch

**可用来创建文件**

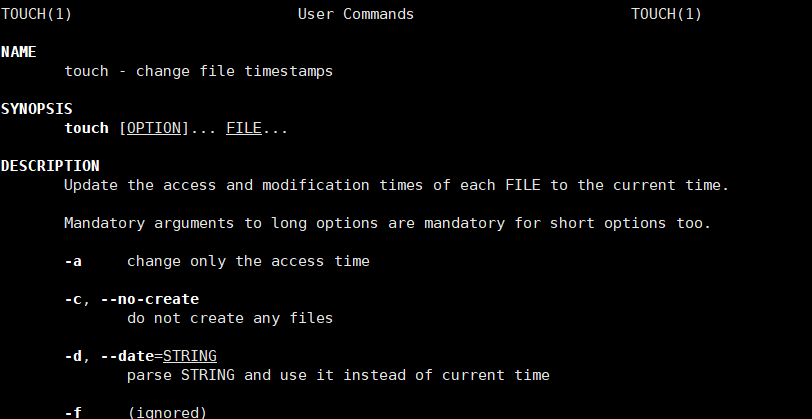
[hc@localhost ~]$ touch a

[hc@localhost ~]$ ls

a Desktop

man touch

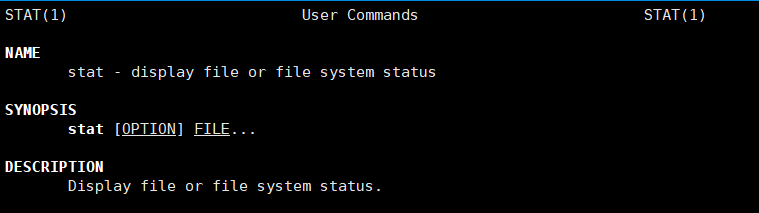
**touch**： **主要是用来改变文件的时间戳**。一个文件有3个时间戳(最近修改，访问，改变)时间，



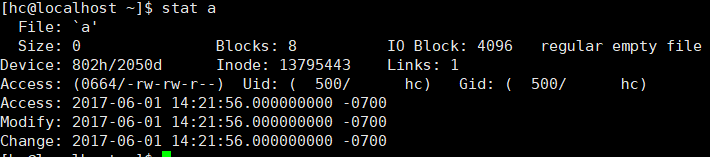
怎么查看一个文件的状态信息呢?

## stat

display file or file system status（显示文件或者文件系统的状态）



stat a

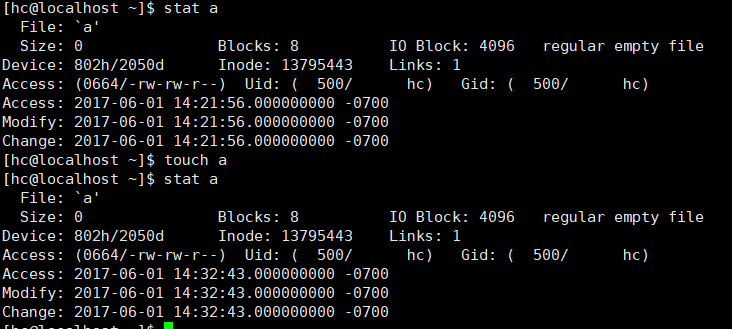


accss: 访问时间

modify:修改时间

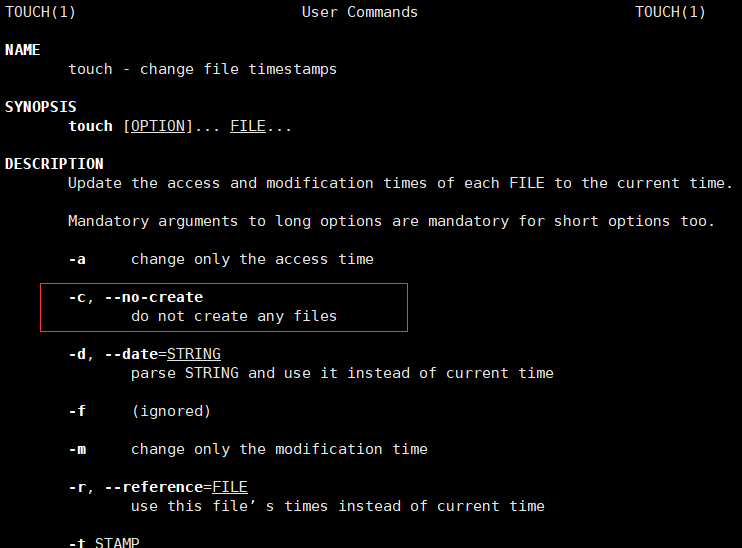
change:改变时间

touch a 后时间改变了 如下图：14：21：56 touch后 14：32：43



**-c : 就不会创建文件**

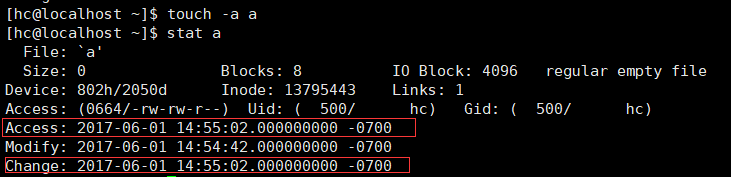
**–c** --no-create do not create any files （不创建任何文件）



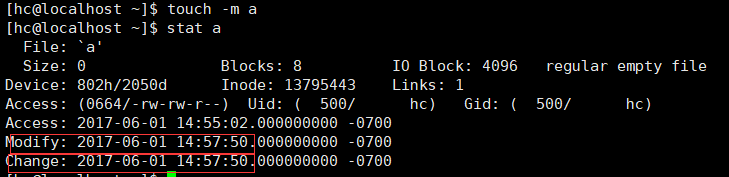
touch -c tst

**-a** change only the access time 只改变访问时间

touch –a a 访问时间改变了



**-m** change only the modification time 只改变修改时间



改变时间 只要touch 都要随之改变。

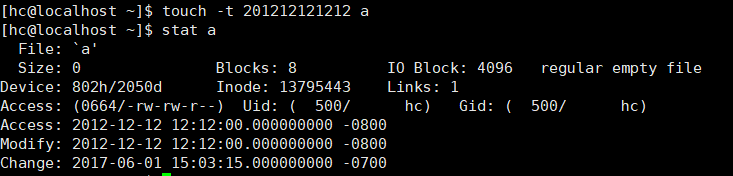
修改为指定时间

**-t STAMP**

use [[CC]YY]MMDDhhmm[.ss] instead of current time

**-d, --date=STRING**

parse STRING and use it instead of current time



创建文件还可以使用文件编辑器：

ASCII 美国信息交换标准代码 128个字符，

2^6 = 0 ~32

2^7 = 0~127

0 000 000 – 1 111 111

计算机中存储数据最小单位是字节，

一个字节8bit, 0000 0000 – 1111 1111 这个区间来表示

ASCII 对其进行了规定，所以是纯文本编辑器

word不是文本编辑器而是字处理器

汉字：也就10多万个，所以 2^16 = 0~65535 ， utf8 2个字节表示英文，3个字节表示中文。

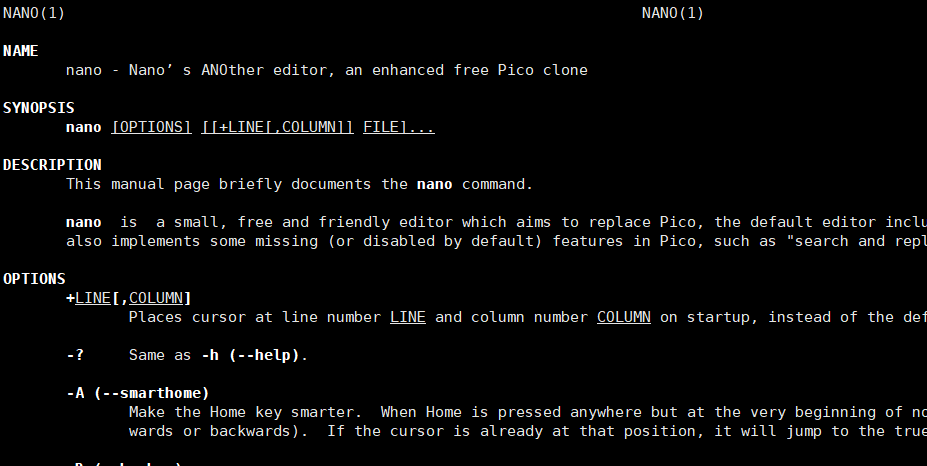
那么系统怎样识别是英文字母还是汉子呢，这就取决与软件的字符集。

国标： GB18030 ,GBK ,GB2312, unicode(全球化字符集编码 4个字节 32bit)

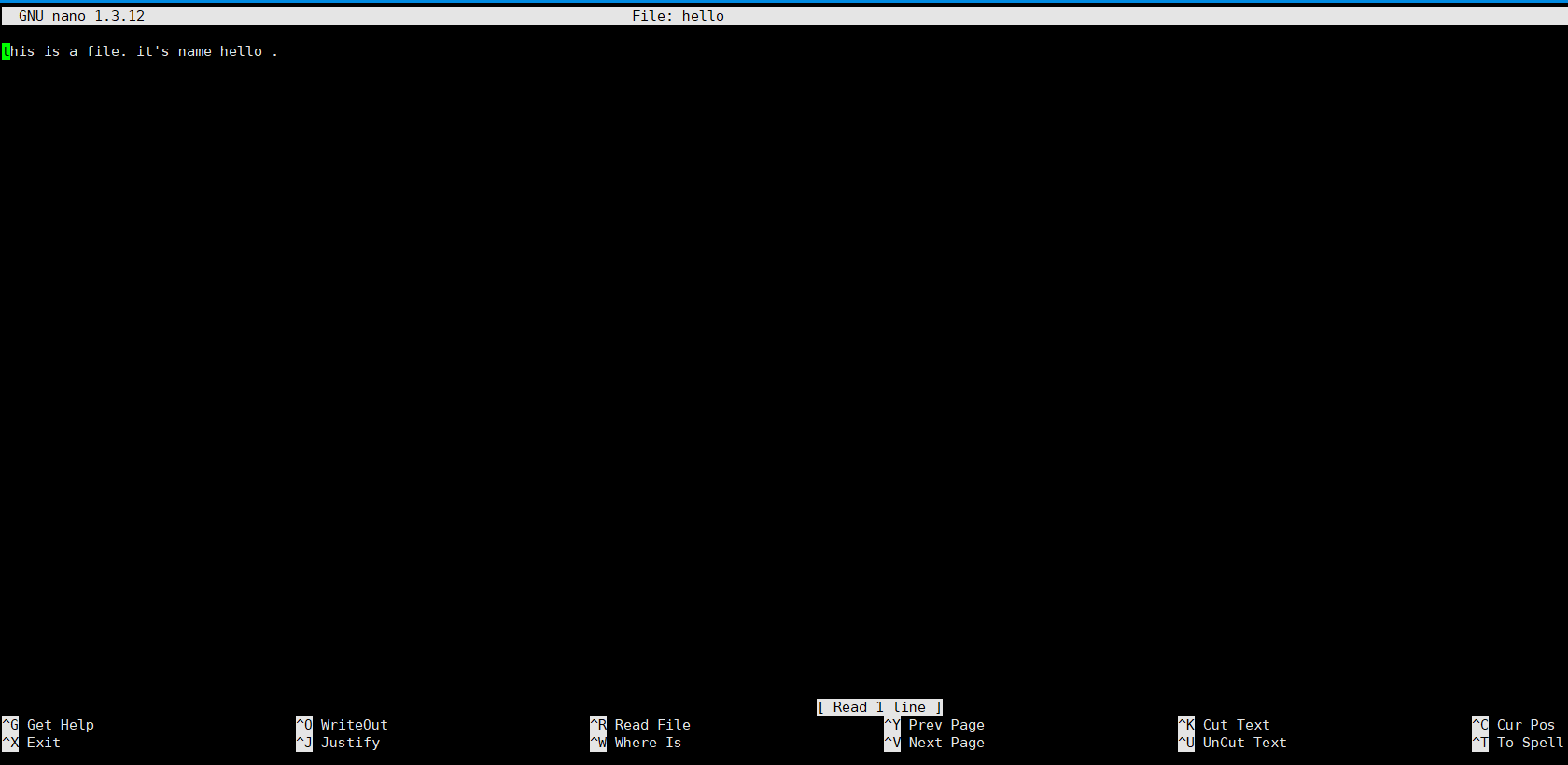
## nano

Linux中的一个纯文本编辑器

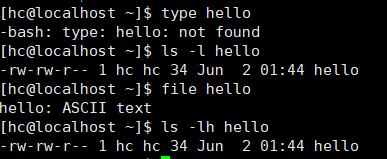
**nano a**



nano : 是一个文本编辑器

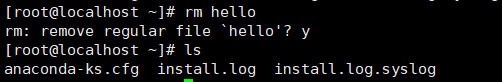


^ : 托字符，就是ctr ctr+o保存 ctr+x 退出



## rm

**删除文件**



**-i : interactive 交互式的，一旦删除的时候给你交互一下，问你要不要删?**

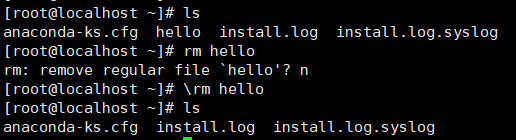
remove regular fiel ‘hello’? 确定删除文件hello吗？这里为什么会提示，我们看一下rm

type rm: rm is aliased to ‘rm -i’ rm是rm -i的别名

默认情况下，普通用户创建一个文件，然后删除，是不会和你交互，不会提示你要不要删除，只有管理员有这么一个选项。为了避免误操作。

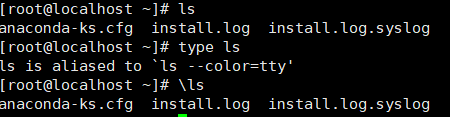


当然管理员要强行删除也可以：



**\rm hello 就不会提示。**

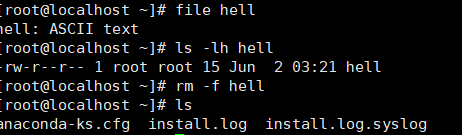
**\ 转义，针对有别名的命令。就会使用原来的命令.**



ls is aliased to ‘ls --color=tty’ ls默认使用 ls –color = tty 默认在终端用颜色来显示文件。

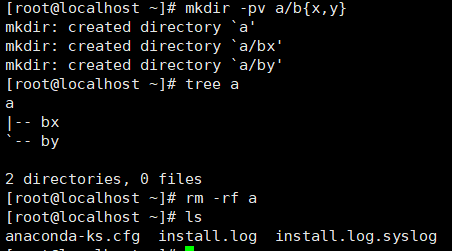
/ls = l s != ( ls –color = tty )

**-f :** force 强制 如果加 -f 就不会给提示，直接删除。



**-r：** recursive（递归） 可以递归删除文件夹

**rm -rf : 递归强制删除目录和文件还不提示**



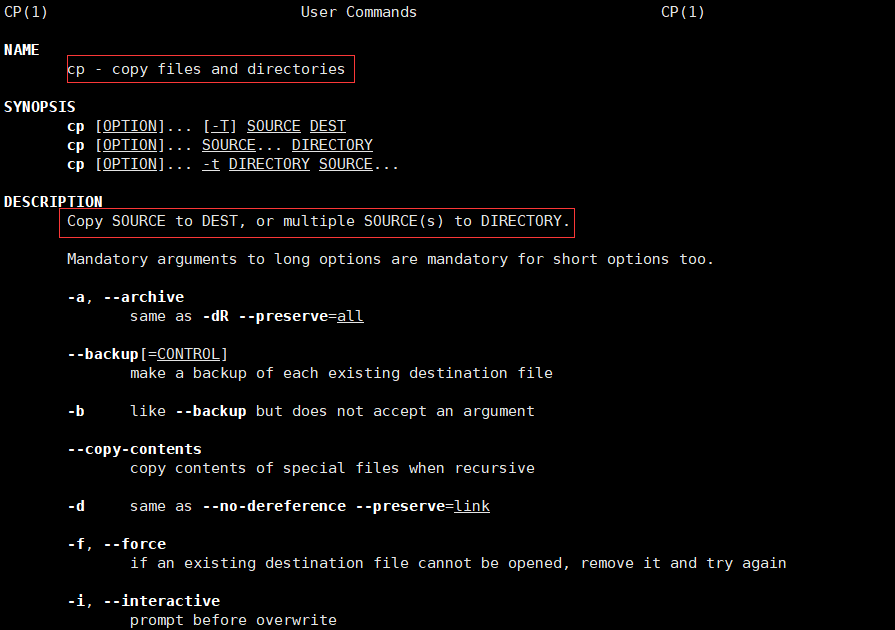
**大招： rm -rf / 删除根目录下的所有文件，绝对致命，绝对有效，绝对没商量！！！！！！！！**

**重要法则：**

**rm -rf 这个命令对于管理员来说实在是太危险了，所以都不用管理员登陆，进行文件操作。他的权限太大了，在实际生产环境当中，最好在删除前做备份。在删除文件时要得到确定和许可才能删除。**

## cp

**copy 复制文件和目录**



cp source to dest or multiple source(s) to dirctory

cp f1 f2 f3 f1和f2 是源文件，f3是目标,反正最后一个才是目标

cp 可以多个源但是目标只有一个，

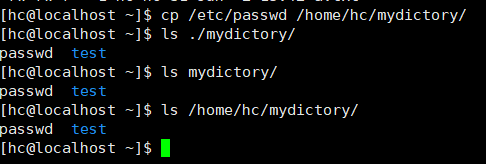
一个文件到一个文件，多个文件到一个目录。

如果源是多个，那么目标只能是目录。

不能多个文件复制为一个文件。

[hc@localhost ~]$ cp /etc/passwd mydictory/ = cp /etc/passwd ./mydictory/ 相对路径

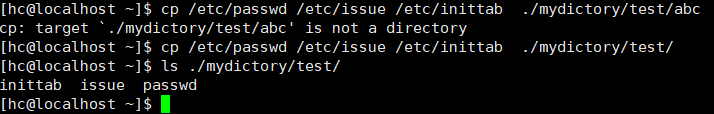
[hc@localhost ~]$ cp /etc/passwd /home/hc/mydictory/ 绝对路径



cp /etc/passwd mydictory/ 将 etc目录下的passwd文件 cp 到mydictory目录下并保存文件名为passwd

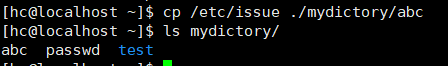
cp /etc/passwd /etc/issue /etc/inittab ./mydictory/test/abc 这里要将etc目录下的3个文件cp到test目录下并保存为abc,结果说 target abc 不是一个目录。说明cp多个源文件，目标必须是一个目录，而不是一个文件。

cp /etc/passwd /etc/issue /etc/inittab ./mydictory/test/ 结果成功



如果源文件是一个，就可以保存

**cp /etc/issue ./mydictory/abc** 就可以将etc目录下的issue文件 cp 到mydictory中并更名为abc

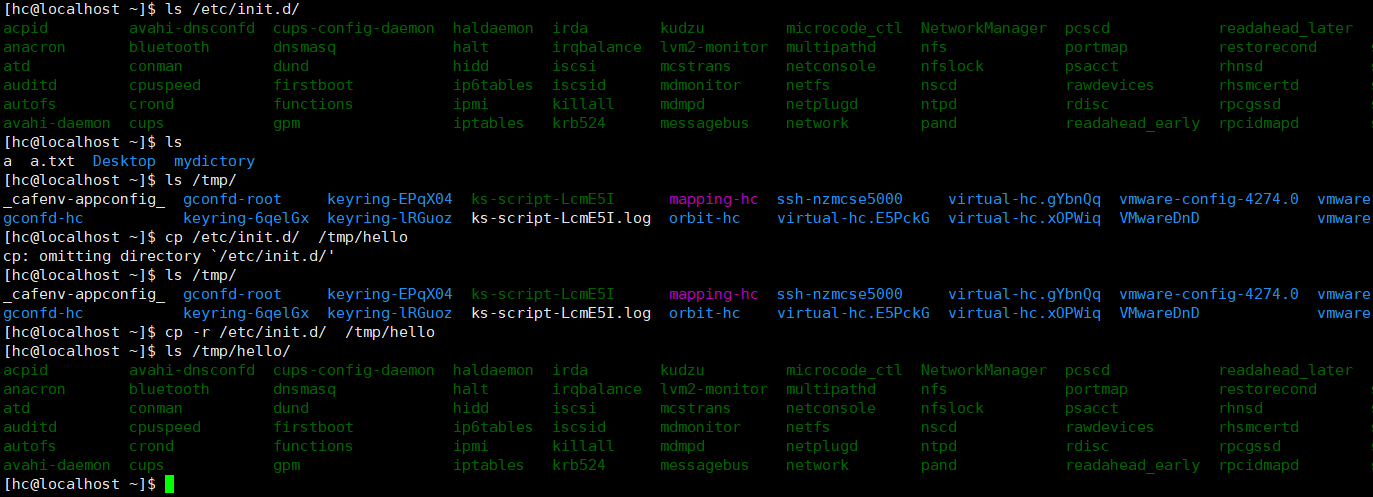


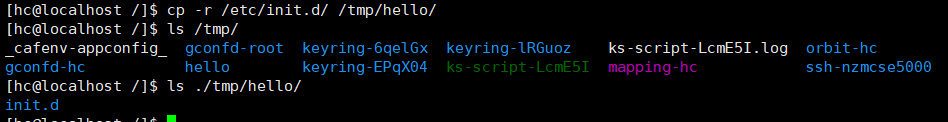
cp 默认情况下只复制文件，不会复制目录。

**-r or -R or --recursive**: **recursive** (递归复制，包括目录和文件)

cp /etc/init.d/ /tmp/hello 将etc目录下的init.d目录下的所有文件递归 cp 到tmp下的hello , 原来hello 是不存在的，这里是init.d目录变为了hello ，如果原来hello目录存在则直接放进去,目录结构 /tmp/hello/init.d/…

如果原来hello 是文件保存为一个链接文件，这是不对的。

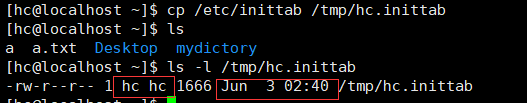


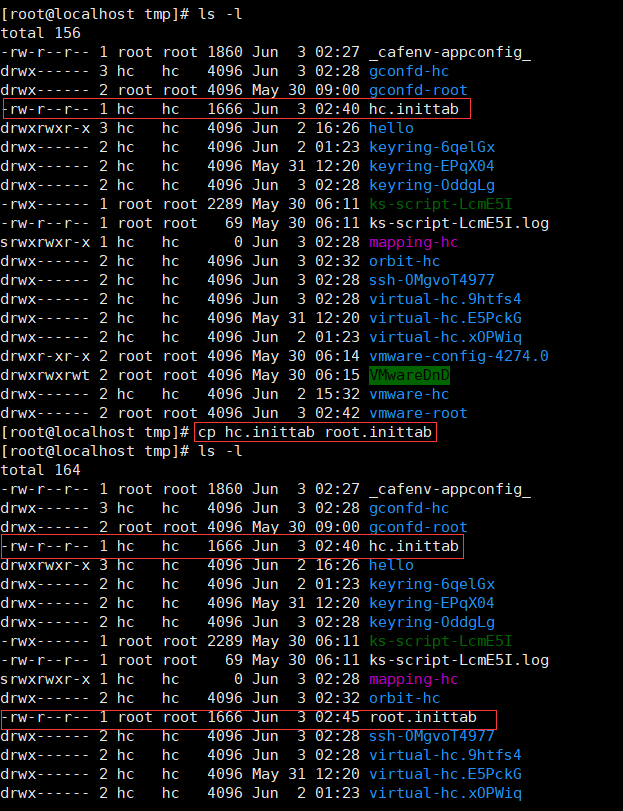


**-f** : **force 强制复制，如果目标目录中有这个文件，强制覆盖，反正无论如何要复制过去。**

**-i ： --interactive 复制前交互式**

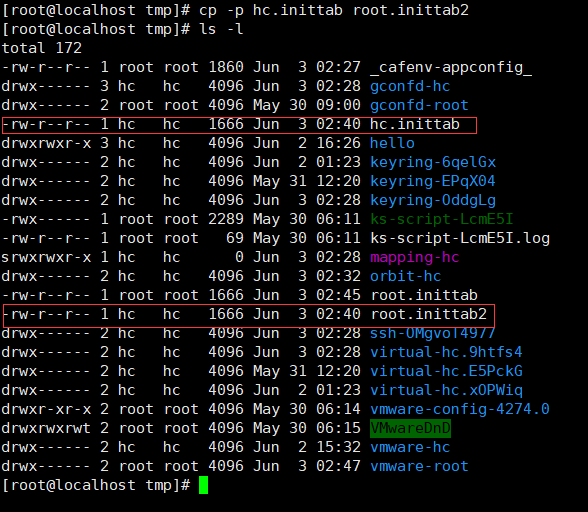
**文件的复制，谁复制的这个文件就属于谁的**。



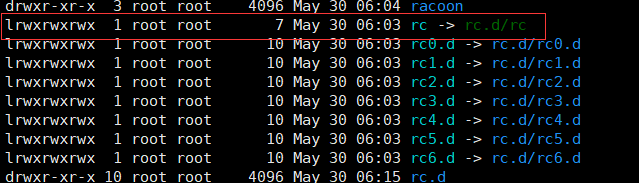


**-p : same as --preserve=mode,ownership,timestamps**

**-p 就和 --preserve一样，保留文件原来的mode（权限），ownership(属组，属主)， timestamps(时间戳)。**

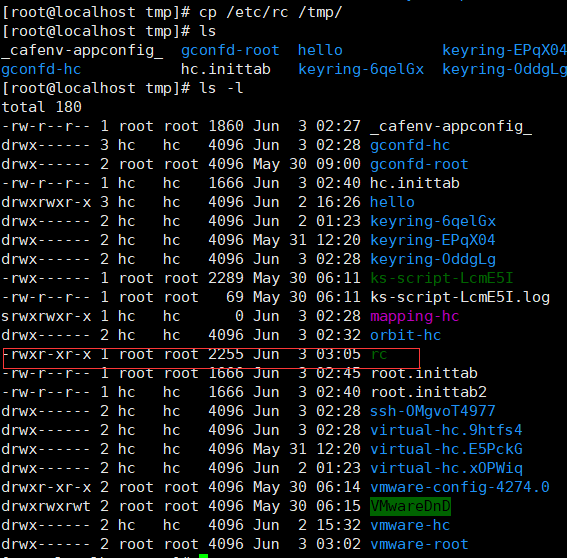


ls -l /etc/



如： rc 就是一个快捷键，它指向rc.d/rc

cp /etc/rc /tmp/



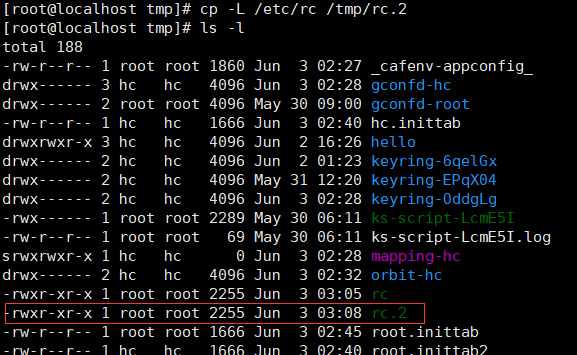
rc就不是一个链接文件了，是一个普通文件。所有复制链接默认是复制的链接所指向的那个文件。而不是链接本身。

**-L : --dereference(废弃，不保存链接自身，默认使用，和上面不写一样的效果)**

**always follow symbolic links(总是跟随符号链接，就是Linux中的快捷键方式，指向 另一个文件，复制链接所指向的文件)**

cp -L /etc/rc /tmp/rc.2

**复制的是文件而不是链接**。

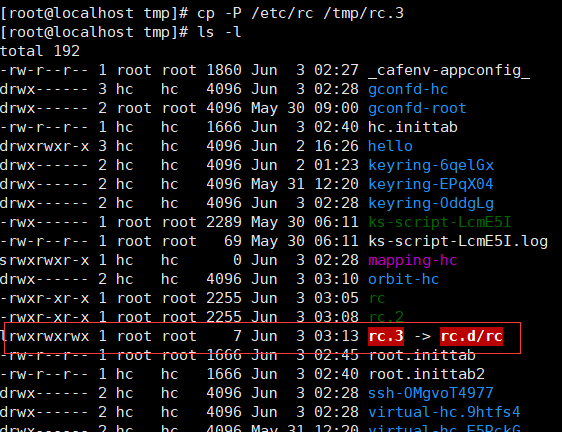


**-P : --no-dereference （保存链接自身）**

**never follow symbolic links（不跟随符号链接，就是复制链接自身）**

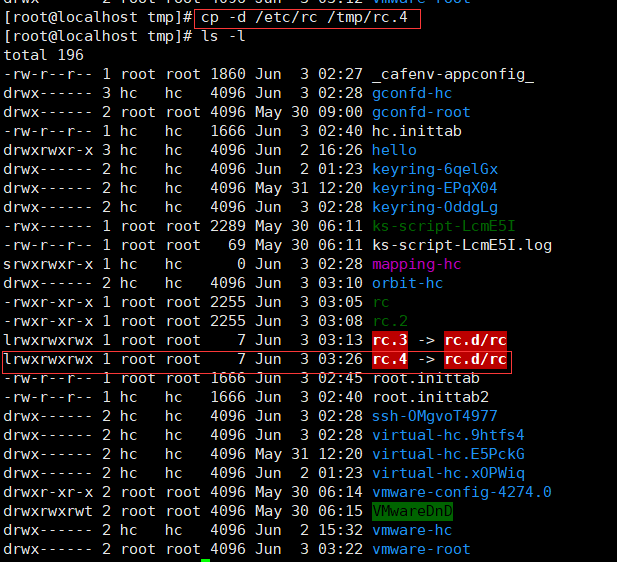
**cp -P /etc/rc /tmp/rc.3**

**如果是一个链接文件，cp后也是一个链接文件。**



**-d** ：**same as --no-dereference --preserve=link**

**和 -p 一样 --preserve=link 保持链接自身**



**-R, -r, --recursive(递归复制)**

**copy directories recursively**

**-a, --archive same as -dR --preserve=all**

**archive（归档存放，备份常用），和 -dR --preserve**

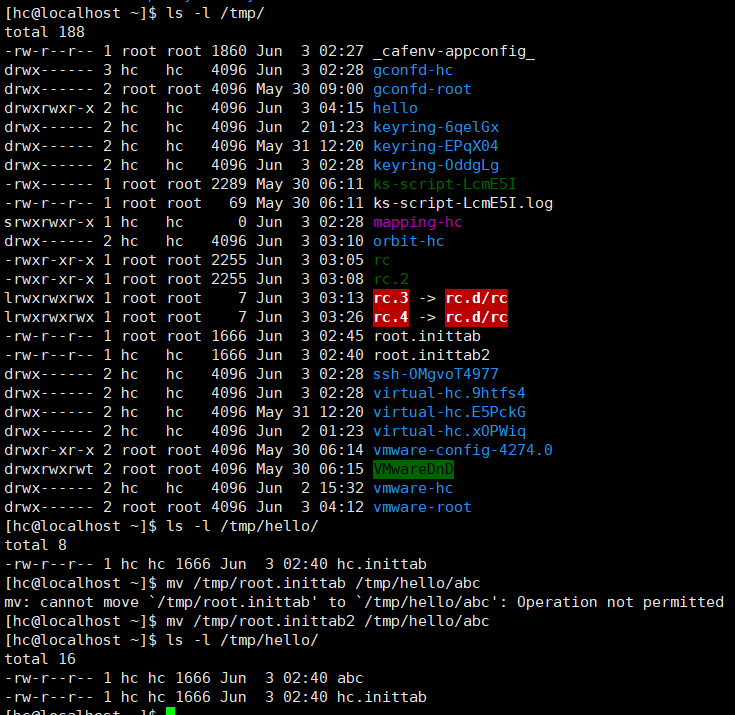
**保留文件原来所有的属性。保留的属性比 -p 还多，cp -a 常用于备份。**

**cp /etc/{inittab,passwd,rc.d/rc.sysinit} /tmp** : 用命令行展开，复制3个文件到tmp目录下

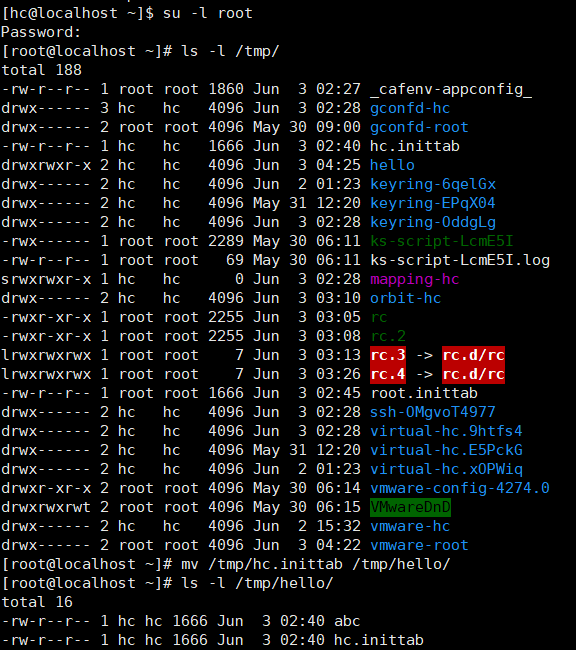
## mv

move 移动文件相当于Windows中的剪切功能

mv /tmp/root.inittab /tmp/hello/abc 将tmp目录下的root.inittab 移动到tmp目录下 的hello目录并命名为abc，当然，用户只能移动自己文件。不能移动他人的文件



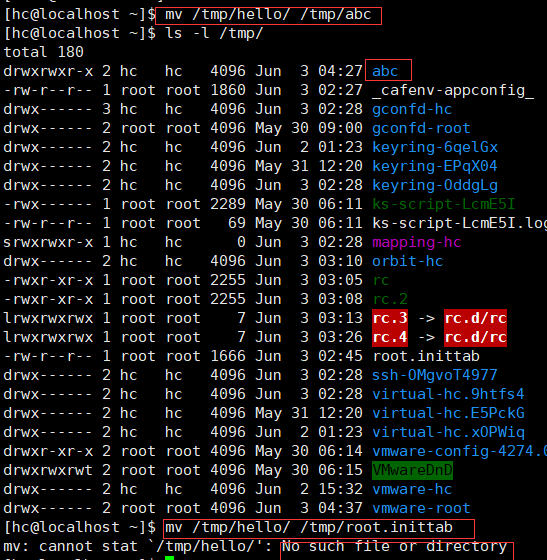
管理员可以移动他人文件



如果源是一个目录，目标是一个文件，

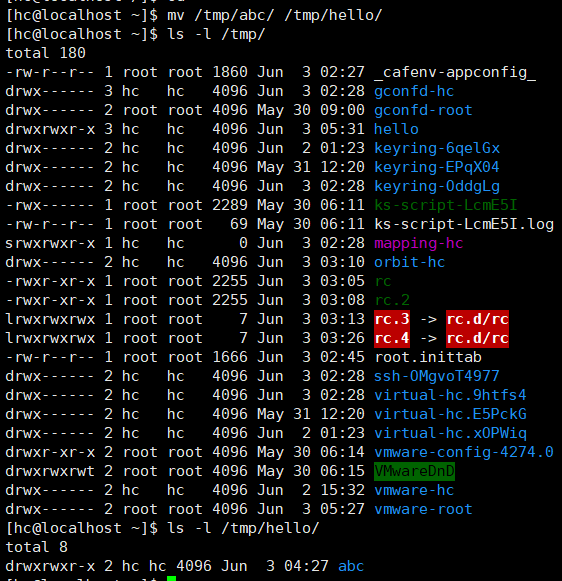
mv /tmp/hello/ /tmp/abc 如果tmp目录不存在abc 那么将hello目录改名为abc目录存放。移动并重命名。

mv /tmp/hello/ /tmp/root.inittab 如果目标是一个已存在文件，那么就会出错，不能用一个目录去覆盖一个非目录。



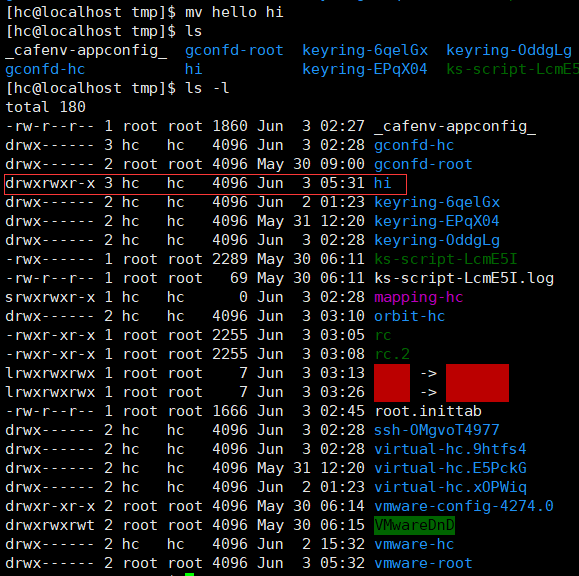
**cp** **mv** 不能多个目标。可以一个或多个源。这是基本法则。

mv /tmp/abc/ /tmp/hello hello 是一个已存在目录，那么就会将abc目录移动到hello目录下。

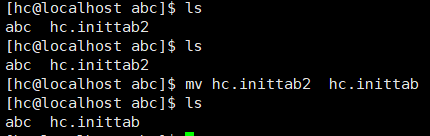


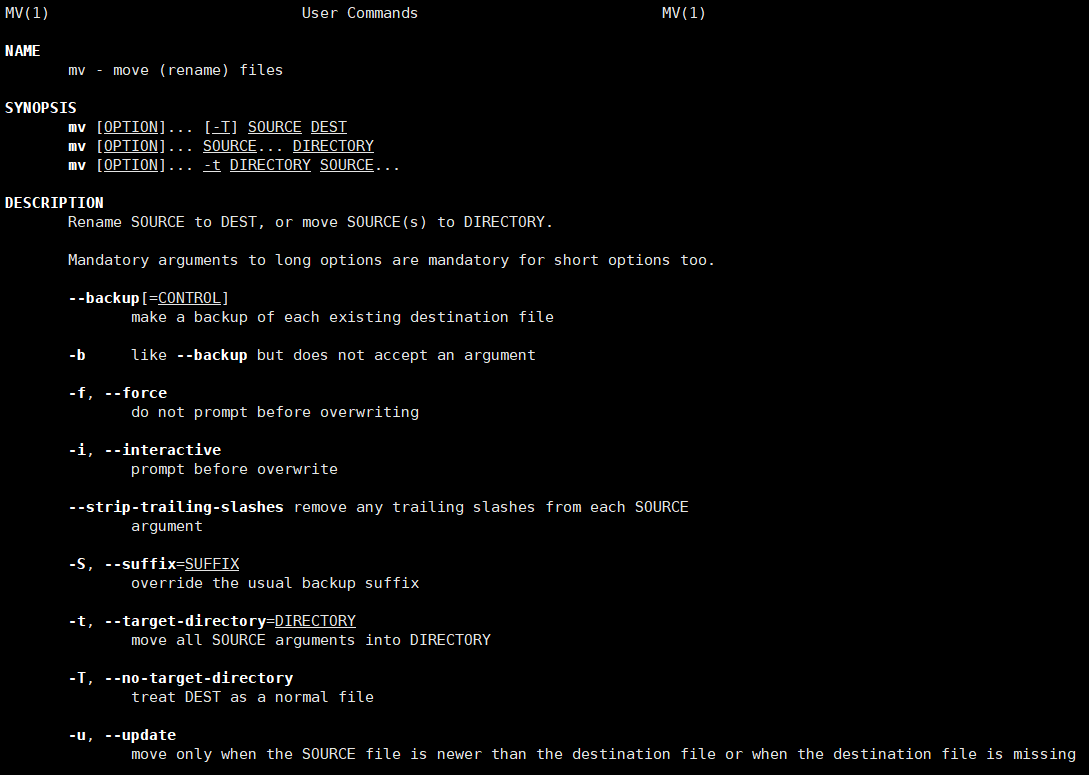
在/tmp 下 使用相对路径。

[hc@localhost tmp]$ mv hello hi 重命名hello目录 为hi目录



[hc@localhost abc]$ mv hc.inittab2 hc.inittab 重命名hc.inittab2 文件为hc.inittab文件





**-f** : --force（强制）

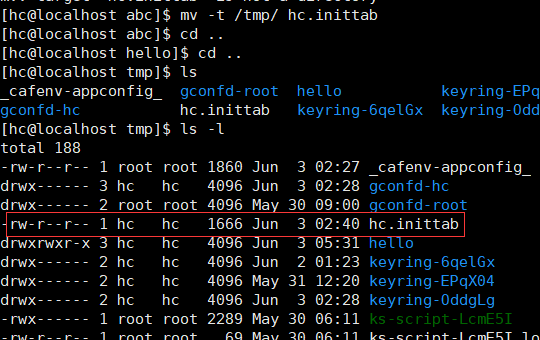
do not prompt before overwriting 强制覆盖，在覆盖之前不要提示

**-t** : --target-directory=DIRECTORY

move all SOURCE arguments into DIRECTORY

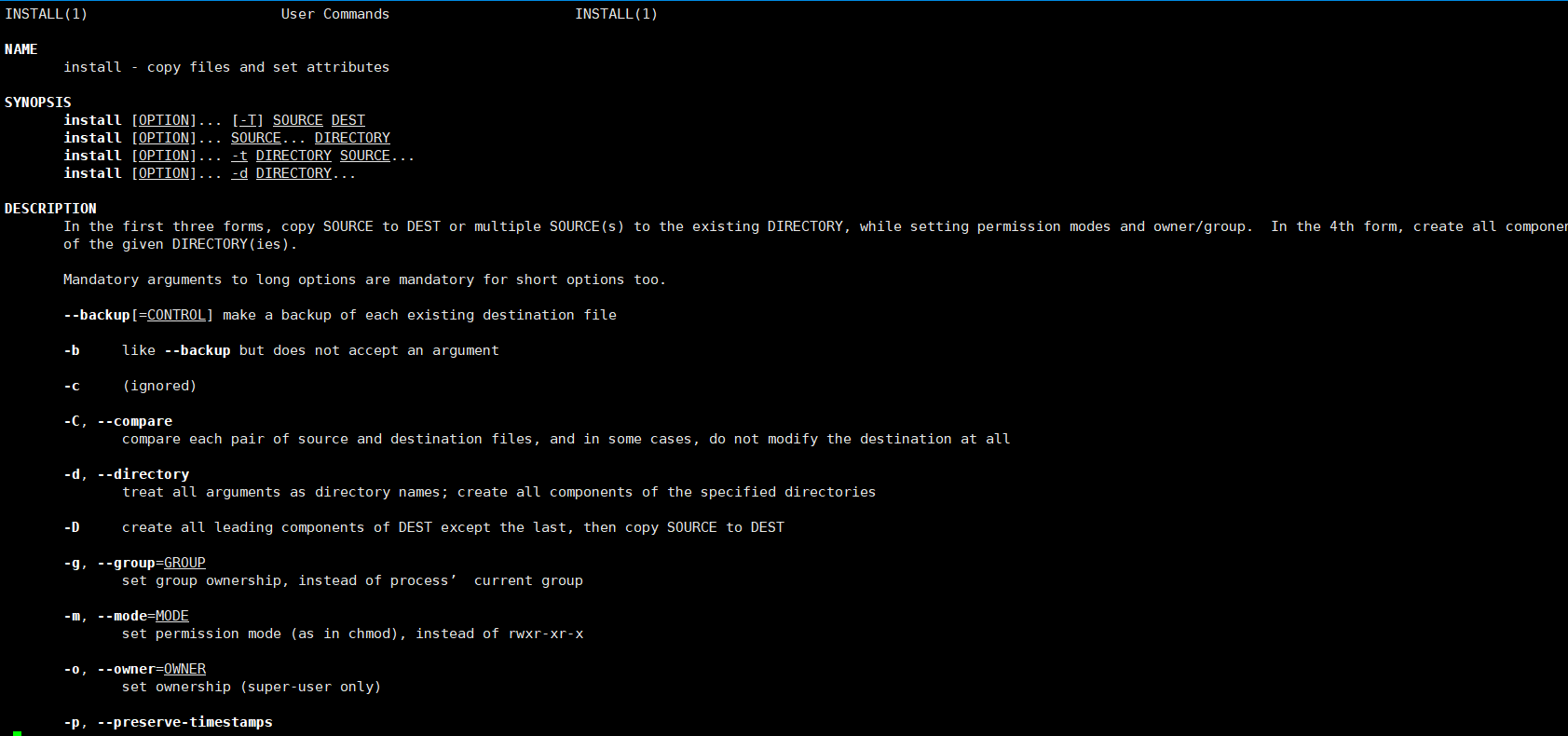
mv -t dest source ,如果 -t 先指定目标，在指定源。

[hc@localhost abc]$ mv -t /tmp/ hc.inittab



## install

**install - copy files and set attributes 拷贝文件并设置属性**



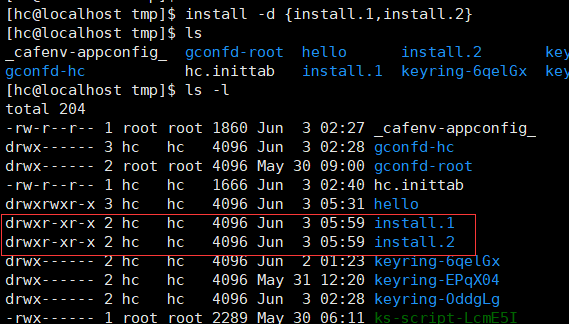
**-d :** --directory

treat all arguments as directory names; create all components of the specified directories

**(**将所有参数视为目录名;创建指定目录的所有组件)

[hc@localhost tmp]$ install -d {install.1,install.2}

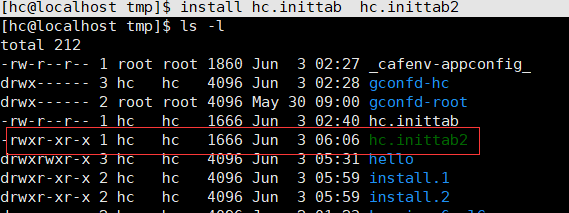
在/tmp目录下创建两个目录install.1,install.2



当然也可以用来复制文件

[hc@localhost tmp]$ install hc.inittab hc.inittab2

将tmp目录下的hc.inittab 复制为hc.inittab2



但是hc.inittab2是绿色的，

hc.inittab 的权限是rw-r--r--

hc.inittab2 的权限是 rwxr-xr-x 表明是多了执行权限。

**-m** : --mode=MODE

set permission mode (as in chmod), instead of rwxr-xr-x

(可以设置权限，默认rwxr-xr-x,有执行权限)

**-t** :--target-directory=DIRECTORY

copy all SOURCE arguments into DIRECTORY

(复制所有的源到目录，就是把多个文件（不能是目录）复制到同一个目录中去)

