<https://blog.csdn.net/lwnylslwnyls/article/details/7902994>

[**cp**](http://www.linuxso.com/command/cp.html) (复制档案或目录)  
[root@linux ~]# **cp** [-a[df](http://www.linuxso.com/command/df.html)ilpr[su](http://www.linuxso.com/command/su.html)] 来源档(source) 目的檔(destination)  
[root@linux ~]# **cp** [options] source1 source2 source3 …. directory  
**参数：**  
-a ：相当于 -pdr 的意思；  
-d ：若来源文件为连结文件的属性(link [file](http://www.linuxso.com/command/file.html))，则复制连结文件属性而非档案本身；  
-f ：为强制 (force) 的意思，若有重复或其它疑问时，不会询问使用者，而强制复制；  
-i ：若目的檔(destination)已经存在时，在覆盖时会先询问是否真的动作！  
-l ：进行硬式连结 (hard link) 的连结档建立，而非复制档案本身；  
-p ：连同档案的属性一起复制过去，而非使用预设属性；  
-r ：递归持续复制，用于目录的复制行为；  
-s ：复制成为符号连结文件 (symbolic link)，亦即『快捷方式』档案；  
-u ：若 destination 比 source 旧才更新 destination ！  
最后需要注意的，如果来源档有两个以上，则最后一个目的文件一定要是『目录』才行！

范例：  
**范例一：将家目录下的 .bashrc 复制到 /tmp 下，并更名为 bashrc**  
[root@linux ~]# [cd](http://www.linuxso.com/command/cd.html) /tmp  
[root@linux tmp]# cp ~/.bashrc bashrc  
[root@linux tmp]# cp -i ~/.bashrc bashrc  
cp: over[write](http://www.linuxso.com/command/write.html) `basrhc’? n  
# 重复作两次动作，由于 /tmp 底下已经存在 bashrc 了，加上 -i 参数，  
# 则在覆盖前会询问使用者是否确定！可以按下 n 或者 y 呢！  
# 但是，反过来说，如果不想要询问时，则加上 -f 这个参数来强制直接覆盖！

**范例二：将 /var/log/wtmp 复制到 /tmp 底下**  
[root@linux tmp]# cp /var/log/wtmp . <==想要复制到目前的目录，最后的 . 不要忘  
[root@linux tmp]# [ls](http://www.linuxso.com/command/ls.html" \t "_blank) -l /var/log/wtmp wtmp  
-rw-rw-r– 1 root utmp 71808 Jul 18 12:46 /var/log/wtmp  
-rw-r–r– 1 root root 71808 Jul 18 21:58 wtmp  
# 注意到了吗？！在不加任何参数的情况下，档案的所属者会改变，连权限也跟着改变了～  
# 这是个很重要的特性！要注意喔！还有，连档案建立的时间也不一样了！  
# 如果您想要将档案的所有特性都一起复制过来，可以加上 -a 喔！  
[root@linux tmp]# cp -a /var/log/wtmp wtmp\_2  
[root@linux tmp]# ls -l /var/log/wtmp wtmp\_2  
-rw-rw-r– 1 root utmp 71808 Jul 18 12:46 /var/log/wtmp  
-rw-rw-r– 1 root utmp 71808 Jul 18 12:46 wtmp\_2  
# 瞭了吧！整个资料特性完全一模一样ㄟ！真是不赖～这就是 -a 的特性！

**范例三：复制 /etc/ 这个目录下的所有内容到 /tmp 底下**  
[root@linux tmp]# cp /etc/ /tmp  
cp: omitting directory `/etc’ <== 如果是目录，不能直接复制，要加上 -r 的参数  
[root@linux tmp]# cp -r /etc/ /tmp  
# 还是要再次的强调喔！ -r 是可以复制目录，但是，档案与目录的权限会被改变～  
# 所以，也可以利用 cp -a /etc /tmp 来下达指令喔！

**范例四：将范例一复制的 bashrc 建立一个连结档 (symbolic link)**  
[root@linux tmp]# ls -l bashrc  
-rw-r–r– 1 root root 395 Jul 18 22:08 bashrc  
[root@linux tmp]# cp -s bashrc bashrc\_slink  
[root@linux tmp]# cp -l bashrc bashrc\_hlink  
[root@linux tmp]# ls -l bashrc\*  
-rw-r–r– 2 root root 395 Jul 18 22:08 bashrc  
-rw-r–r– 2 root root 395 Jul 18 22:08 bashrc\_hlink  
lrwxrwxrwx 1 root root 6 Jul 18 22:31 bashrc\_slink -> bashrc  
# 那个 bashrc\_slink 是由 -s 的参数造成的，建立的是一个『快捷方式』，  
# 所以您会看到在档案的最右边，会显示这个档案是『连结』到哪里去的！  
# 至于那个 bashrc\_hlink 有趣了！建立了这个档案之后， bashrc 与 bashrc\_hlink  
# 所有的参数都一样，只是，第二栏的 link 数改变成为 2 了～而不是原本的 1 喔！  
# 这两种连结的方式的异同，我们会在下一章里面进行介绍的！

**范例五：若 ~/.bashrc 比 /tmp/bashrc 新才复制过来**  
[root@linux tmp]# cp -u ~/.bashrc /tmp/bashrc  
# 这个 -u 的特性，是在目标档案与来源档案有差异时，才会复制的。  
# 所以，比较常被用于『备份』的工作当中喔！ ^\_^

**范例六：将范例四造成的 bashrc\_slink 复制成为 bashrc\_slink\_2**  
[root@linux tmp]# cp bashrc\_slink bashrc\_slink\_2  
[root@linux tmp]# ls -l bashrc\_slink\*  
lrwxrwxrwx 1 root root 6 Jul 18 22:31 bashrc\_slink -> bashrc  
-rw-r–r– 1 root root 395 Jul 18 22:48 bashrc\_slink\_2  
# 这个例子也是很有趣喔！原本复制的是连结档，但是却将连结档的实际档案复制过来了  
# 也就是说，**如果没有加上任何参数时，复制的是源文件，而非连结文件的属性**！  
# 若要复制连结文件的属性，就得要使用 -d 或者 -a 的参数了！

**范例七：将家目录的 .bashrc 及 .bash\_history 复制到 /tmp 底下**  
[root@linux tmp]# cp ~/.bashrc ~/.bash\_history /tmp  
# 可以将多个数据一次复制到同一个目录去！  
这个 cp 的功能很多，而由于我们常常在进行一些数据的复制，所以也会常常用到这个指令的。 一般来说，我们如果去复制别人的数据 (当然，该档案您必须要有 read 的权限才行啊！ ^\_^) 时， 总是希望复制到的数据最后是我们自己的，所以，在预设的条件中， cp 的来源档与目的档的权限是不同的，目的档的拥有者通常会是指令操作者本身。举例来说， 上面的范例二中，由于我是 root 的身份，因此复制过来的档案拥有者与群组就改变成为 root 所有了！ 这样说，可以明白吗？！ ^\_^

由于具有这个特性，因此，当我们在进行备份的时候，某些需要特别注意的特殊权限档案， 例如密码文件 (/etc/shadow) 以及一些设定档，就不能直接以 cp 来复制，而必须要加上 -a 或者是 -p 等等可以完整复制档案权限的参数才行！另外，如果您想要复制档案给其它的使用者，也必须要注意到档案的权限(包含读、写、执行以及档案拥有者等等)，否则，其它人还是无法针对您给予的档案进行修订的动作喔！注意注意！

至于上面的范例当中，第四个范例是最有趣的，使用 -l 及 -s 都会建立所谓的连结档 (link file)，但是这两种连结档确有不一样的展现情况。这是怎么一回事啊？ 那个 -l 就是所谓的 hard link ，至于 -s 则是 symbolic link ，[鸟哥](http://www.linuxso.com/linuxrumen/631.html" \t "_blank)这里先不介绍， 因为这个涉及 i-node 的相关知识，我们还没有介绍到，下一章再来讨论这个 link 的问题喔！ 总之，由于 cp 有种种的档案属性与权限的特性，所以，在复制时，您必须要清楚的了解到：  
• 是否需要完整的保留来源档案的信息？  
• 来源档案是否为连结档 (symbolic link file)？  
• 来源档是否为特殊的档案，例如 FIFO, socket 等？  
• 来源文件是否为目录？

需要说明的是，为防止用户在不经意的情况下用cp[命令](http://www.linuxso.com/command/" \t "_blank)破坏另一个文件，如用户指定的目标文件名已存在，用cp命令拷贝文件后，这个文件就会被新源文件覆盖，因此，建议用户在使用cp命令拷贝文件时，最好使用i选项。

原创地址：http://www.linuxso.com/command/cp.html