<https://blog.csdn.net/arthur_02_13/article/details/53436728>

本文列出了一些在Linux终端上使用频率比较高的一些操作命令。

**1.控制光标**

up(方向键上) 可以调出输入历史执行记录，快速执行命令

down(方向键下) 配合 up 选择历史执行记录

Home 移动光标到本行开头

End 移动光标到本行结尾

PgUp 向上翻页

PaDN 向下翻页

ctrl + c 终止当前程序

Tab 补全，Tab 补全是非常有用的一个功能，可以用来自动补全命令或文件名，省时准确。未输入状态下连按两次 Tab 列出所有可用命令已输入部分命令名或文件名，按 Tab 进行自动补全。

**2.常用命令**

（1）**cd** 是打开某个路径的命令，也就是打开某个文件夹，并跳转到该处。

$ cd path ### path 为你要打开的路径。   
其中 path 有绝对路径和相对路径之分，绝对路径强调从 / 起，一直到所在路径。相对路径则相对于当前路径来说，假设当前家目录有etc 文件夹（绝对路径应为 /home/username/etc），如果直接 cd etc 则进入此文件夹，但若是 cd /etc/ 则是进入系统 etc ，多琢磨一下就可以理解了。另外在 Linux 中， . 代表当前目录， .. 代表上级目录，因此返回上级目录可以 cd .. 。

（2）**ls** 即 list ，列出文件。注意： Linux 中 以 . 开头的文件或文件夹均为隐藏文件或隐藏文件夹。

$ ls ### 仅列出当前目录可见文件

$ ls -l ### 列出当前目录可见文件详细信息

$ ls -hl ### 列出详细信息并以可读大小显示文件大小

$ ls -al ### 列出所有文件（包括隐藏）的详细信息

$ls -lt ###对最后修改日期和时间进行排序 基于日期和时间的逆向排序文件

$ls -ltr ### 基于日期和时间的逆向排序文件

（3）**pwd** 用于返回当前工作目录的名字，为绝对路径名。

$ pwd   
/home   
mkdir

（4）**mkdir** 用于新建文件夹。

$ mkdir folder

$ mkdir -p folder/subfolder ### -p 参数为当父目录存在时忽略，若不存在则建立，用此参数可建立多级文件夹

（5）**rm** 即 remove ，删除文件。

$ rm filename ### 删除 filename

$ rm -i filename ### 删除 filename 前提示，若多个文件则每次提示

$ rm -rf folder/subfolder/ ### 递归删除 subfolder 下所有文件及文件夹，包括 subfolder 自身

$ rm -d folder ### 删除空文件夹

$ rm -v !(“filename1”|”filename2”|\*.zip) ### 删除某文件外的所有文件

（6）**cp** 即 copy ，复制文件。

$ cp source dest ### 将 source 复制到 dest

$ cp folder/\* dest ### 将 folder 下所有文件(不含子文件夹中的文件)复制到   
dest

$ cp -r folder dest ### 将 folder 下所有文件（包含子文件夹中的所有文件）复制到 dest

（7）**mv** 即 move ，移动文件。

$ mv source folder ### 将 source 移动到 folder 下，完成后则为 folder/source

$ mv -i source folder ### 在移动时，若文件已存在则提示 **是否覆盖**

$ mv source dest ### 在 dest 不为目录的前提下，重命名 source 为 dest

（8）**cat** 用于输出文件内容到 Terminal 。

$ cat /etc/locale.gen ### 输出 locale.gen 的内容

$ cat -n /etc/locale.gen ### 输出 locale.gen 的内容并显示行号

（9）**more**

more 与 cat 相似，都可以查看文件内容，所不同的是，当一个文档太长时， cat 只能展示最后布满屏幕的内容，前面的内容是不可见的。这时候可用 more 逐行显示内容。

more/etc/locale.genmore/etc/locale.gen more +100 /etc/locale.gen ### 从 100 行开始显示   
less

（11）**less** 与 more 相似，不过 less 支持上下滚动查看内容，而 more 只支持逐行显示。

less/etc/locale.genless/etc/locale.gen less +100 /etc/locale.gen

（12）**nano** 是一个简单实用的文本编辑器，使用简单。

$ nano filename ### 编辑 filename 文件，若文件不存在，则新打开一个文件，若退出时保存，则创建该文件。编辑完后，ctrl + X 提示是否保存，按 y 确定保存即可。

注意：在使用过程中可用 ctrl + G 获取帮助。

（13）**reboot**

reboot 为重启命令。

reboot ‘$’ 普通用户即可执行 而 ‘#’ 为 root 用户才可执行，或普通用户使用 ‘sudo’

（14）**poweroff**

poweroff 为关机命令。

（15）**ping**

ping 主要用于测试网络连通，通过对目标机器发送数据包来测试两台主机是否连通，及延时情况。

$ ping locez.com ### 通过域名 ping，若 DNS 未设置好，可能无法 ping 通

$ ping linux.cn

PING linux.cn (211.157.2.94) 56(84) bytes of data.

64 bytes from 211.157.2.94.static.in-addr.arpa (211.157.2.94): icmp\_seq=1 ttl=53 time=41.5 ms

64 bytes from 211.157.2.94.static.in-addr.arpa (211.157.2.94): icmp\_seq=2 ttl=53 time=40.4 ms

64 bytes from 211.157.2.94.static.in-addr.arpa (211.157.2.94): icmp\_seq=3 ttl=53 time=41.9 ms   
^C   
— linux.cn ping statistics —   
3 packets transmitted, 3 received, 0% packet loss, time 2002ms

rtt min/avg/max/mdev = 40.406/41.287/41.931/0.644 ms

$ ping 211.157.2.94 ### 通过 IP 地址 ping ，若无法 ping 通可能是网络连接出现问题

（16）**grep**

grep 主要用于返回匹配的项目，支持正则表达式。

$ grep PATTERN filename ### 返回所有含有 PATTERN 的行

$ grep zh\_CN /etc/locale.gen ### 返回所有含 zh\_CN 的行

（17）**mount**

mount 用于挂载一个文件系统，需要 root 用户执行。一个磁盘可分为若干个分区，在分区上面可以创建文件系统，而挂载点则是提供一个访问的入口，将一个分区的文件系统挂载到某个目录中，称这个目录为挂载点，并且可以通过这个挂载点访问该文件系统中的内容。

例如一块硬盘在 Linux 中表示为 /dev/sda 那么它上面的分区应该表示为 /dev/sda1 、/dev/sda2 。

（18）**umount**

umount 与 mount 相反，是卸载一个挂载点，即取消该入口。

（19）**tar**

tar 主要用于创建归档文件，和解压归档文件，其本身是没有压缩功能的，但可以调用 gzip 、 bzip2 进行压缩处理。   
参数解释：

-c 创建归档   
-x 解压归档   
-v 显示处理过程   
-f 目标文件，其后必须紧跟 目标文件   
-j 调用 bzip2 进行解压缩   
-z 调用 gzip 进行解压缩   
-t 列出归档中的文件   
$ tar -cvf filename.tar . ### 将当前目录所有文件归档，但不压缩，注意后面有个 ’.‘ ，不可省略，代表当前目录的意思

$ tar -xvf filename.tar ### 解压 filename.tar 到当前文件夹

$ tar -cvjf filename.tar.bz2 . ### 使用 bzip2 压缩

$ tar -xvjf filename.tar.bz2 ### 解压 filename.tar.bz2 到当前文件夹

$ tar -cvzf filename.tar.gz ### 使用 gzip 压缩

$ tar -xvzf filename.tar.gz ### 解压 filename.tar.gz 到当前文件夹

$ tar -tf filename ### 只查看 filename 归档中的文件，不解压

（20）**ln**

ln 主要用于在两个文件中创建链接，链接又分为 Hard Links (硬链接)和 Symbolic Links (符号链接或软链接)，其中默认为创建硬链接，使用 -s 参数指定创建软链接。

硬链接主要是增加一个文件的链接数，只要该文件的链接数不为 0 ，该文件就不会被物理删除，所以删除一个具有多个硬链接数的文件，必须删除所有它的硬链接才可删除。

软链接简单来说是为文件创建了一个类似快捷方式的东西，通过该链接可以访问文件，修改文件，但不会增加该文件的链接数，删除一个软链接并不会删除源文件，即使源文件被删除，软链接也存在，当重新创建一个同名的源文件，该软链接则指向新创建的文件。

硬链接只可链接两个文件，不可链接目录，而软链接可链接目录，所以软链接是非常灵活的。

$ ln source dest ### 为 source 创建一个名为 dest 的硬链接

$ ln -s source dest ### 为 source 创建一个名为 dest 的软链接

（21）**chown**

chown 用于改变一个文件的所有者及所在的组。

（22）**chmod**

chmod 永远更改一个文件的权限，主要有 读取 (4)、 写入(2) 、 执行(1) ，三种权限，其中 所有者 、 用户组 、 其他 各占三个，因此 ls -l 可以看到如下的信息

-rwxr–r– 1 locez users 154 Aug 30 18:09 filename   
其中 r=read ， w=write ， x=execute

（23）**useradd**

useradd 用于添加一个普通用户。

（24）**passwd**

passwd 用于改变用户登录密码。

$ passwd ### 不带参数更改当前用户密码

（25）**whereis**

whereis 用于查找文件、手册等。

$ whereis bash

bash: /usr/bin/bash /etc/bash.bashrc /etc/bash.bash\_logout /usr/share/man/man1/bash.1.gz /usr/share/info/bash.info.gz

$ whereis -b bash ### 仅查找 binary   
bash: /usr/bin/bash /etc/bash.bashrc /etc/bash.bash\_logout

$ whereis -m bash ### 仅查找 manual   
bash: /usr/share/man/man1/bash.1.gz /usr/share/info/bash.info.gz   
find

find 也用于查找文件，但更为强大，支持正则，并且可将查找结果传递到其他命令。

$ find . -name PATTERN ### 从当前目录查找符合 PATTERN 的文件

$ find /home -name PATTERN -exec ls -l {} \; # 从 /home 文件查找所有符合 PATTERN 的文件，并交由 ls 输出详细信息

（26）**wget**

wget 是一个下载工具，简单强大。

$ wget -O newname.md   
<https://github.com/LCTT/TranslateProject/blob/master/README.md>   
### 下载 README 文件并重命名为 newname.md

$ wget -c url ### 下载 url 并开启断点续传

（27）iconv -l 所有已有编码的字符集

iconv -f ISO-8859-1 -t UTF-8//TRANSLIT input.file -o out.file 更改文件的编码格式

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/Arthur\_02\_13/article/details/53436728