

Linux 常用命令介绍

【命令格式】**ls** [参数]

【命令作用】查看当前目录下所有子文件的名字

【命令示例】**ls -a**

【参数说明】**-a** 察看所有子文件, 包括隐藏文件
-c 以更改时间排序, 显示文件和目录
-d 仅仅列出目录本身

【命令格式】**cd** [file_holder_name]

【命令作用】打开一个文件夹

【命令示例】**cd hello** 打开一个名为 hello 文件夹
cd .. 打开父目录
cd ~ 打开自己用户所在的 home directory

【参数说明】无

【Linux 技巧】可以先用 **ls** 查看文件夹下的子文件, 然后再使用 **cd** 打开某个子文件夹。注意, 这个命令只能用于文件夹哦。在 **ubuntu** 中, 文件夹的名字会高亮。

【命令格式】**cp** [参数] [source file name] [destination]

【命令作用】将 source file 复制到 destination

【命令示例】**cp ~/.bashrc /tmp/bashrc**

【参数说明】**-r** 递归持续复制, 用于整个目录的复制行为
-i 如果目标文件已经存在, 会在覆盖的时候给予询问

【命令格式】**rm** [参数] [delete file name]

【命令作用】将文件删除

【命令示例】**rm /tmp/bashrc**

【参数说明】**-f** 如果文件是不存在的, 不会给出提醒
-i 删除前会对用户进行询问
-r 递归地删除, 适用于删除文件夹, 但小心使用这个参数

【命令格式】**mv** [参数] [source file name] [destination]

【命令作用】将文件移动到 destination

【命令示例】**mv /tmp/bashrc /home**

【参数说明】**-f** 如果目标文件已经存在, 会直接覆盖
-i 如果目标文件已经存在, 会对用户进行询问是否覆盖
-u 如果目标文件已经存在, 而且 source 比较新, 那么才会更新 destination

【Linux 技巧】

- 1) 如果你想重命名某个文件, 你可以这样写 **mv a.txt b.txt**, 这样文件 **a.txt** 就变成了 **b.txt**
- 2) 如果你想移动一批类似的文件, 如后缀名相同, 你可以写 **mv *.txt /home/git**, 这样所有 **.txt** 的文件就都会被移动到 **/home/git** 里面去了。

【命令格式】**mkdir** [file_holder_name]
【命令作用】在当前文件夹创建一个文件夹
【命令示例】**mkdir hello**
【参数说明】无

【命令格式】**sudo** [other linux command]
【命令作用】以管理员身份执行命令
【命令示例】**sudo mkdir hello**
【说明】大部分命令之前加上 **sudo** 有可以以管理员身份执行。我们只有在自己用户的 home directory（也就是在~中）中执行命令的时候不需要其他权限，否则一律需要以管理员身份执行，例如：在 /usr 文件夹中执行命令等。

【命令格式】**tar** [参数] [-f filename] [被压缩、解压的目录]
【命令作用】进行压缩、解压
【命令示例】**tar -jcv -f filename.tar.bz2 tar_file**
 （对 tar_file 压缩）
tar -jxv -f filename.tar.bz2 -C hello
 （对 filename.tar.bz2 进行解压，并将解压结果放在 hello 文件夹中）
【参数说明】-c 新建打包文件
 -x 解压缩功能
 -j 支持 bzip2 的压缩，此时文件后缀名应该是 .tar.bz2
 -z 支持 gzip 的压缩，此时文件后缀名应该是 .tar.gz
 -C 当解压缩的时候，如果想在特定目录下解压缩，可以适用这个参数

【命令格式】**chmod** [参数] [mode] [files]
【命令作用】改变 files 的权限
【命令示例】**chmod u+x file**
【参数说明】mode 可以由 who opcode permission 三部分组成
 Who 可以是：u（当前用户）、g（当前组）、o（其它）；这个是可选得，默认是所有用户。
 Opcode 可以是：+（增加权限）、-（删除权限）、=（重新分配权限）
 Permission 可以是：r（读）、w（写）、x（执行）
 作为选择，我们多数用三位八进制数字的形式来表示权限，第一位指定属主的权限，第二位指定组权限，第三位指定其他用户的权限，每位通过 4(读)、2(写)、1(执行)三种数值的和来确定权限。如 6(4+2)代表有读写权，7(4+2+1)有读、写和执行的权限。

 --help: 输出帮助信息。
 -R, --recursive: 可递归遍历子目录，把修改应到目录下所有文件和子目录

【命令格式】**cat** [**参数**] [**filename**]

【命令作用】察看文件内容

【命令示例】**cat** hello.txt

【参数说明】**-E**, **--show-ends** 在每行结束处显示 **\$**
-n, **--number** 对输出的所有行编号
-s, **--squeeze-blank** 不输出多行空行

【命令格式】**man** [**linux command**]

【命令作用】查询你不了解得 linux 命令

【命令示例】**man** ls

【参数说明】无

【文件路径表示】

../ 表示上一级目录，相对路径写法

~ 用户私有目录

./ 当前目录

G++参数

-Wall 一般使用该选项，允许发出 GCC 能够提供的所有有用的警告。也可以用 **-W{warning}** 来标记指定的警告。

-c 只激活预处理,编译,和汇编,也就是说使用这个参数，G++只会把代码编译成 **obj** 文件，不会做连接处理。

-g 只是编译器，在编译的时候，产生调试信息。

-o 制定目标名称, 缺省的时候, gcc 编译出来的文件是 **a.out**

例子: **gcc -o hello.p hello.c**