

Linux基本操作

一、目录操作

- **cd**

用法: `cd [选项] 目录`

作用: 切换到某个目录 (Change Directory) 。

<code>cd /</code>	进入根目录
<code>cd ~</code>	进入当前用户主目录(/home/用户名)
<code>cd -</code>	返回上次所在的目录
<code>cd .</code>	当前目录
<code>cd ..</code>	返回上一级目录
<code>cd /etc</code>	进入/etc目录(绝对路径)
<code>cd etc</code>	进入/etc目录(相对路径)
<code>cd etc/</code>	进入/etc目录(相对路径)

Tips: 在需要键入文件名或目录名时, 可以使用 Tab 自动补足全名, 当有多种补全方案时双击 Tab 可以显示所有可能选项。

- **ls**

用法: `ls [选项] [文件或目录]`

作用: 列出目录中的文件。若参数“目录”未给出, 则列出当前目录中的文件。

<code>ls -l</code>	每行只列出一个文件
<code>ls -a</code>	列出所有文件, 包括隐藏文件

- **pwd**

用法: `pwd [选项]`

作用: 显示当前目录的绝对路径。

- **mkdir**

用法: `mkdir [选项] 目录`

作用: 创建目录。

- **rmdir**

用法: `rmdir` [选项] 目录
作用: 删除空目录。请注意, 只有空的目录才能被删除

二、文件操作

• touch

用法: `touch` [选项] 文件
作用: 当文件存在时更新文件的时间戳, 当文件不存在时创建新文件。

```
touch helloworld.c
```

• rm

用法: `rm` [选项] 文件
作用: 删除文件。

选项 (常用):

<code>-r</code>	递归删除目录及其内容, 删除非空目录必须有此选项。
<code>-f</code>	强制删除, 不提示用户确认, 忽略不存在的目录。
<code>-i</code>	逐一提示用户确认每个将要被删除的文件。

Tips: `rm -rf` 是十分危险的命令 (尤其在 `root` 用户下), 非必要不使用 `rm -rf` 命令, 在执行之前需要再三确认。`root` 用户具有至高无上的权限, 在该用户下执行 `rm -rf /` 可以删除一切文件, 包括 Linux 本身, 从而导致系统被毁灭。

• cp

用法: `cp` [选项] 源文件 目标路径
作用: 将源文件 (也可以是目录) 复制为目标路径对应的文件 (如果目标路径是文件) 或复制到目标路径 (如果目标路径是目录)。

选项 (常用):

<code>-r</code>	递归复制目录及其子目录内的所有内容。
-----------------	--------------------

• mv

用法: `mv` [选项] 源文件 目标路径
作用: 将源文件 (也可以是目录) 移动为目标路径对应的文件 (如果目标路径是文件) 或移动到目标路径 (如果目标路径是目录)。

选项 (常用):

<code>-v</code>	显示详细的操作信息。
-----------------	------------

-i	进行交互式操作，在覆盖前询问。
-u	仅在源文件较新，或目标文件不存在时，才执行移动操作。

当然，mv 命令还可以用于移动目录，不过需要注意的是，与 cp 命令不同，使用 mv 命令移动目录时，不需要添加 -r 选项，它可以直接完成对目录及其子内容的移动。

mv 命令的另外一个作用就是重命名文件。原理上很好理解，也就是把一个文件以不同的名字移动到当前目录下，这和重命名是等价的。举个例子，我们在 dir 下使用 mv test.txt test2.txt 就可以把 test.txt 重命名为 test2.txt 了。

- **diff**

用法：diff [选项] 文件1 文件2 (纯文本文件)

选项（常用）：

-b	不检查空白字符的不同。
-B	不检查空行。
-q	仅显示有无差异，不显示详细信息。

三、查找操作

- **find**

用法：find [路径] <选项>

作用：在给定路径下递归地查找文件，输出符合要求的文件的路径。如果没有给定路径，则在当前目录下查找。

选项（常用）：

-name <文件名>	指定需要查找的文件名
-------------	------------

使用 find 命令并加上 -name 选项可以在当前目录下递归地查找符合参数所示文件名的文件，并将文件的路径输出至屏幕上。

- **grep**

用法：grep [选项] PATTERN FILE

(PATTERN是匹配字符串，FILE是文件或目录的路径)

作用：输出匹配PATTERN的文件和相关的行。

选项（常用）：

-a	不忽略二进制数据进行搜索。
-i	忽略大小写差异。
-r	从目录中递归查找。
-n	显示行号。

- **tree**

用法: tree [选项] [目录名]




选项(常用):

- a 列出全部文件（包含隱藏文件）。
- d 只列出目錄。

开源文本编辑器Vim

1. 创建文件。利用之前学到的 `touch` 命令创建 `helloworld.c` 文件。
2. 打开文件。在创建 `helloworld.c` 文件的目录下，输入 `vim helloworld.c` 打开刚刚新建的文件。
3. 输入内容。刚打开文件时，我们无法向其中直接输入代码。我们需要按 `I` 键进入“插入模式”，之后便可以向其中输入文本内容。如下图所示。

```
#include <stdio.h>
int main() {
    printf("Hello World!\n");
    return 0;
}
```

4. 保存并退出 Vim。完成文件修改后，按  `Esc` 回到命令模式，再按下  `Shift` +  `;` 进入底线命令模式，此时可以看到屏幕的左下角出现了一个冒号，输入 `w`，并按下回车，文件便得到了保存；再进入底线命令模式，输入 `q` 便可以关闭文件，回到命令行界面。其中 `w` 和 `q` 可以合并，也就是输入 `:wq` 也可以完成上面的操作。

常用快捷键

常用的快捷键

- `⌘ Ctrl + C` 终止当前程序的执行。当程序卡死或者你不想让它继续执行了，可以输入 `⌘ Ctrl + C` 终止程序。请注意，`⌘ Ctrl + C` 在 Windows 中是复制的意思，但在 Linux CLI 的环境下就是结束进程的意思。所以在使用终端模拟器（比如使用浏览器或 XShell 等）连接终端时，复制终端中显示的内容时记得不要按 `⌘ Ctrl + C`，而是老老实实右键复制，以免误杀进程。
- `⌘ Ctrl + Z` 挂起当前程序。暂停程序，放到后台。`⌘ Ctrl + Z` 挂起程序后会显示该程序挂起编号，若想要恢复该程序可以使用 `fg [job_spec]` 即可，`job_spec` 即为挂起编号，不输入时默认为最近挂起进程。
- `⌘ Ctrl + D` 终止输入（若正在使用 Shell，则退出当前 Shell）。在标准输入中输入 `⌘ Ctrl + D` 也意味着输入了一个 EOF。
- `⌘ Ctrl + L` 清屏。相当于命令 `clear`。

特别提示，`⌘ Ctrl + S` 在终端中的作用是暂停该终端。有的同学在进行编辑的时候会误触此组合键导致终端“卡死”，此时使用 `⌘ Ctrl + Q` 组合键即可让终端继续运行。

Tips：在多数 Shell 中，四个方向键也是有各自特定的功能的：`← Left` 和 `→ Right` 可以控制光标的位置，`↑ Up` 和 `↓ Down` 可以切换最近使用过的命令。