

(vim) 多文件

手稿



封面

vim 可以同时编辑多个文件，在它们之间自由切换但不用退出。

概述

无论有多少文件，vim 都可以在不退出的情况下对它们进行编辑。你可以在这些文件之间自由切换，还可以来回拷贝文本。

当你准备修改文件时，可以选择保留文件的最初版本以及备份。

编辑其他文件

假设想要在文件 `sailor.c` 中添加一个函数 `coefficient()`，这个函数的声明位于头文件 `sailor.h`，那么就可以用命令 `:edit sailor.h` 打开它。

```
1 // sailor.c: a simple C language example code file.
2 // This is a sample source file. written with vim
3
4 #include <stdio.h>
5
6 double sum(int a, double b, double c)
7 {
8     return a + b + c;
9 }
10
11 int main(int argc, char *argv[])
12 {
13     printf("sum = %f\n", sum(1, 2, 3));
14     return 0;
15 }
```



```
1 // sailor.h: a simple C language example code file.
2 // This is a sample header file. written with vim
3
4 #ifndef SAILOR_H
5 #define SAILOR_H
6
7 double sum(int a, double b, double c);
8
9 #endif
```

`:edit sailor.h`

拷贝文本

来到 `sailor.h` 后，我们可以用命令 `yy` 拷贝（或者在可视化模式下选择后拷贝）函数 `coefficient()` 的原型，然后用命令 `:edit sailor.c` 回到 `sailor.c`，在合适的位置按下 `p` 键放入已拷贝的文本。

```
1 #ifndef SAILOR_H_
2 #define SAILOR_H_
3
4 extern double linear(double a, double b, double x);
5 extern float coefficient(float coe);
6
7 #endif
~
```

第一步：

移至第 5 行

输入 `2yy`，即拷贝 2 行

第二步：

输入 `:edit sailor.c` 返回

```
11 }
12
13 extern float coefficient(float coe);
14
15 int main(int argc, char *argv[])
16 {
```

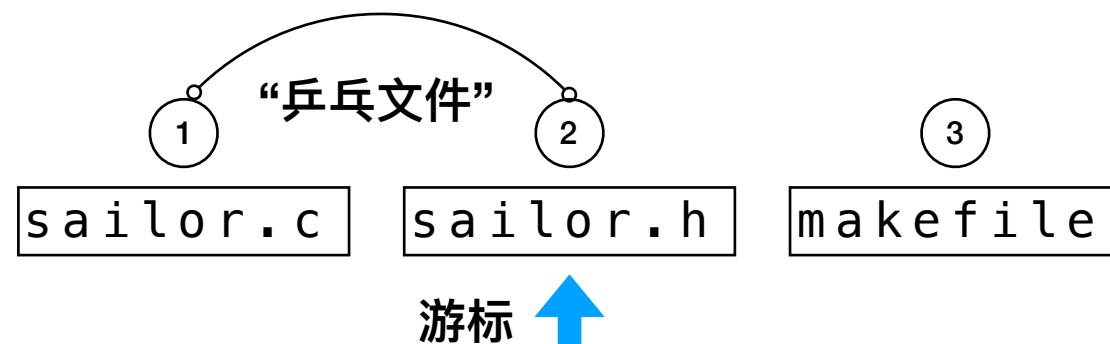
第三步：

在 `main` 前面一行按下 `p` 键

注：用 `:write >> file` 可以将当前文件的全部内容追加到 `file` 末尾。

一次打开多个文件

如果在启动 `vim` 时，后面的参数包含多个文件，那么这些文件就组成了一个待编辑的文件列表。例如：
`$ vim sailor.c sailor.h makefile`
这里有 3 个文件，它们的编号分别是 1、2、3，用游标来指向它们。



`:first` 让游标指向第一个文件
`:last` 让游标指向最后一个文件

初始状态下，`vim` 缓冲区中存放的是编号 1 的 `sailor.c` 文件，游标指向 #1。如果输入 `:next` 命令，`vim` 载入 2 号文件，游标指向 #2。同理，如果再次输入 `:next` 命令则载入 3 号文件，游标指向 #3。

初始状态下，`:2next` 命令让游标指向 #3。

`:previous` 命令与 `:next` 类似，但是在反方向上移动游标。

最近两次使用的文件组成了“乒乓文件”，使用 `C-^` (`Ctrl+Shift+6`) 可以在两者之间快速切换。需要注意的是：初始状态下尚不存在“乒乓文件”；`C-^` 不会改变游标的位置。

哪个文件？

当文件较多时，如果你想知道现在正在编辑的是哪个文件，可以输入命令 `:args`。这时会在最后面一行左下角显示文件列表，正在编辑的文件名用 `[]` 包含。

例如：

```
sailor.c [sailor.h] makefile
```

上面的输出表明正在编辑的文件是 `sailor.h`。

位置记忆

在编辑多个文件时，可以利用下面 2 个预定义标记快速定位。

- `" 带你回到上次离开文件时光标所在位置
- `· 带你回到最后一次修改文件的位置附近

大写标记

在“(vim) 标记”小节中讲到，大写字母标记可用于多个文件之间。
以 `sailor.c`、`sailor.h` 和 `makefile` 为例，我们分别设置如下标记：

sailor.c	6 <code>#include <stdio.h></code>	mC
sailor.h	4 <code>extern double linear(double a, double b, double x);</code>	mH
makefile	1 <code>.PHONY: clean</code>	mM

设置了标记后，我们可以输入 `:marks CHM` 查看以上 3 个标记的详细信息：

```
:marks CHM
mark line col file/text
C      6    0 sailor.c
H      4    0 sailor.h
M      1    0 .PHONY: clean
Press ENTER or type command to continue
```

此后，我们可以用 `'C`、`'H`、`'M` 分别跳转到以上 3 个不同文件的相应位置。

文件备份

正常情况下，`vim` 不使用备份文件。如果你需要这项功能，只需要用 `:set backup` 命令打开备份选项，这样每次对文件做出修改后都会在备份文件中保留前一个版本。

备份文件的名称是原文件名称后跟 `~`，你可以用 `:set backupext` 命令设置其他后缀，例如 `:set backupext=.bak`。

如果你只想备份任何改动之前的文件内容，可以输入 `:set patchmode` 命令，例如 `:set patchmode=.orig`。当第一次修改文件并保存后，`vim` 在当前目录生成一个同名 `.orig` 文件，里面是被编辑文件的初始内容。如果继续修改这个文件，`vim` 会发现已存在一个 `.orig` 文件，因此跳过这一步，继续写入备份文件（`~` 或自定义的后缀文件）。



只读访问

如果你想要以只读方式访问文件，例如 `sailor.c`，可以使用如下命令：

```
$ vim -R sailor.c
```

或者

```
$ view sailor.c
```

或者

```
$ vim -M sailor.c
```