

(vim) 崩溃后恢复

手稿



封面

vim 使用交换文件存放编辑过程中的修改内容，它可用于在编辑会话崩溃后进行恢复。

概述

当正在用 vim 编辑文件时，你是否遇到过下面的情况：

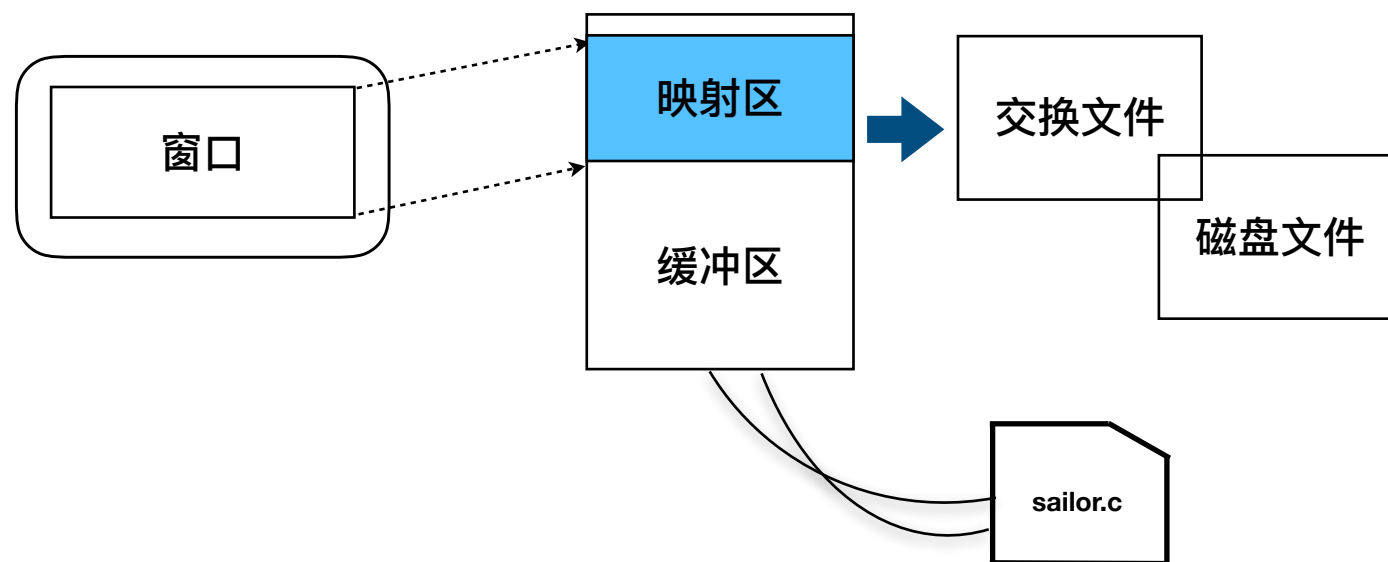
- ◆ 计算机突然崩溃了
- ◆ 虚拟机进入了休眠状态
- ◆ 其他人也在编辑这个文件

上面这些场景均可能导致数据丢失，但大多数情况下，你可以通过交换文件找回丢失的数据。

交换文件

那么什么是交换文件呢？

当你在编辑文件时，`vim` 在交换文件中存放修改的内容。因此，利用初始文件和交换文件，你大概率能够恢复已完成的工作。



交换文件的名字

交换文件的名字通常与编辑文件同名，后跟后缀 `.swp`。在 Unix 系统中，这个文件名前面还要加上一个点（`.`），这样它们平时（用 `ls -a` 命令可以显示 `.` 文件）就不会出现在文件列表中。

当 `.swp` 文件已存在时，`vim` 会给出警告，后来的交换文件则使用后缀 `".sw0"`、`".swn"` 等等，你可以用模式 `*.s[uvw][a-z]` 来匹配交换文件。

交换文件的位置

默认情况下，交换文件位于与编辑文件相同目录下。你可以设置 `directory` 选项来修改交换文件的位置。

当 `vim` 处于命令模式时，可以输入 `:sw` 命令查看当前编辑文件的交换文件。

```
[lettuce@lettuce:~/source/480-vim_base$ vi sailor.h
```

```
:sw
```

```
/home/lettuce/source/480-vim_base/.sailor.h.swp
```

更新交换文件

当你键入 200 字符、或者 4 秒内未键入任何内容时，`vim` 就会更新交换文件。但必须是缓冲区内容被修改了，如果只是移动光标不会更新交换文件。

那么为什么不一直更新交换文件呢？原因是这样做可能会严重拖慢正常工作。

要键入多少字符后更新交换文件可通过 `updatecount` 选项更改；同理，多长时间未键入任何内容更新交换文件可通过 `updatetime` 选项更改，参数单位是毫秒。

当写入交换文件后，`vim` 将文件同步（`sync`）到磁盘。如果正处于一个繁忙的 `Unix` 系统，这一步可能需要一些时间，增加了丢失数据的风险。

提示信息

假设正在编辑 `sailor.h` 文件时，系统出现了意外情况。如果 `vim` 生成了交换文件，在你重新打开这个文件时会遇到类似下面这个提示。

E325: ATTENTION

```
Found a swap file by the name ".sailor.h.swp"
    owned by: lettuce   dated: Fri Jun 24 20:05:31 2022
    file name: ~lettuce/source/480-vim_base/sailor.h
    modified: YES
    user name: lettuce   host name: lettuce
    process ID: 1457
While opening file "sailor.h"
    dated: Thu Jun 23 17:49:45 2022
```

- (1) Another program may be editing the same file. If this is the case, be careful not to end up with two different instances of the same file when making changes. Quit, or continue with caution.
- (2) An edit session for this file crashed.
If this is the case, use `:recover` or `vim -r sailor.h` to recover the changes (see `:help recovery`).
If you did this already, delete the swap file `".sailor.h.swp"` to avoid this message.

```
Swap file ".sailor.h.swp" already exists!
[O]pen Read-Only, (E)dit anyway, (R)ecover, (D)elele it, (Q)uit, (A)bort: █
```

选择项

提示消息中最后一行的选择项包括：

<code>[O]pen Read-Only</code>	以只读方式打开文件
<code>(E)dit anyway</code>	仍然编辑这个文件，要小心使用！
<code>(R)ecover</code>	从交换文件恢复文件
<code>(D)elete it</code>	删除交换文件
<code>(Q)uit</code>	退出，不去编辑文件
<code>(A)bort</code>	退出，同时中断后续命令（例如脚本中编辑多个文件）

恢复文件

你可以在命令行终端输入如下命令来恢复数据：

```
$ vim -r sailor.h
```

如果文件被成功恢复，你会看到关于它的提示消息。

```
Using swap file ".sailor.h.swp"
Original file "~/source/480-vim_base/sailor.h"
Recovery completed. You should check if everything is OK.
(You might want to write out this file under another name
and run diff with the original file to check for changes)
You may want to delete the .swp file now.
```

```
Press ENTER or type command to continue
```

按下回车后，`vim` 显示的就是恢复后的文件内容。

这时你必须仔细检查文件内容，确认无误后再保存、退出，然后删除交换文件。