

第10章 找回文本

在 GNU Emacs 中, 无论你何时用 “kill” 命令从缓冲区中剪切了文本, 你都能用一个 “yank” 命令将其重新找回。从缓冲区中剪切的文本被放在 kill 环中, yank 命令则将 kill 环中适当的内容重新插入到缓冲区中 (不一定非得是原来的缓冲区)。

一个简单的 C-y (yank) 命令, 就从 kill 环中取出第一个元素插入到当前的缓冲区中。如果 C-y 命令后面紧跟一个 M-y 命令, 则不是第一个元素而是第二个元素被插入到当前缓冲区中。连续的 M-y 命令则将使第三个元素或第四个元素等代替第二个元素而被插入到当前缓冲区中。当这样不断键入 M-y 而到达 kill 环的最后一个元素时, 它就循环地将第一个元素插入到当前缓冲区中。(因此 kill 环被称为 “环”, 而不仅仅是列表。然而, 保存文本的实际数据结构是一个列表。关于将一个列表作为环来处理的详细内容, 参见附录B, “处理 kill 环”。)

10.1 kill 环总览

kill 环是文本字符串的一个列表。它类似于下面的列表:

```
("some text" "a different piece of text" "yet more text")
```

如果 kill 环的内容就是这样的一个列表, 当键入 C-y 时, 字符串 “some text” 将插入到当前缓冲区中光标当前所处的位置 (即位点处)。

yank 命令也用于复制文本, 这通过拷贝文本实现。被拷贝的文本不是从缓冲区中删除掉, 但是这部分文本的一个拷贝被放到 kill 环中, 并可以用 yank 命令将其找回来插入到当前缓冲区中。

能够将文本从 kill 环中找回的函数有三个: yank 函数, 通常绑定到 C-y; yank-pop 函数, 通常绑定到 M-y; rotate-yank-pointer 函数, 这个函数被前面两个函数使用。

这些函数通过一个被称为 kill-ring-yank-pointer 的变量指向 kill 环。事实上, yank 函数和 yank-pop 函数中插入文本的代码都是下面这个表达式:

```
(insert (car kill-ring-yank-pointer))
```

在开始理解 yank 函数和 yank-pop 函数如何工作之前, 分析一下 kill-ring-yank-pointer 变量和 rotate-yank-pointer 函数是必要的。

10.2 kill-ring-yank-pointer 变量

就像 kill-ring 是一个变量一样, kill-ring-yank-pointer 也是一个变量。它通过被绑定到相应的值来指向某些东西, 这一点与其他 Lisp 变量没有什么两样。

因而, 如果 kill 环的值是:

```
("some text" "a different piece of text" "yet more text")
```

