# World Hello



2010-12-08

# Vim 复制粘贴探秘

Vim作为最好用的文本编辑器之一，使用vim来编文档，写代码实在是很惬意的事情。每当学会了vim的一个新功能，就会很大地提高工作效率。有人使用vim几十年，还没有完全掌握vim的功能，这也说明了vim的强大。而这样何尝不是一件好事呢，只要有的学习，就有的提高。 最近使用Vim来写博客，发现在Vim中粘贴Python代码后，缩进就全乱了。仔细研究了以下，原来是自动缩进的缘故，于是做如下设置：

```
:set noai nosi
```

取消了自动缩进和智能缩进，这样粘贴就不会错行了。但在有的vim中不行，还是排版错乱。后来发现了更好用的设置：

```
:set paste
```

进入paste模式以后，可以在插入模式下粘贴内容，不会有任何变形。这个真是灰常好用，情不自禁看了一下帮助，发现它做了这么多事：

- textwidth设置为0
- wrapmargin设置为0
- set noai
- set nosi
- softtabstop设置为0
- revins重置
- ruler重置
- showmatch重置
- formatoptions使用空值

下面的选项值不变，但却被禁用：

- lisp
- indentexpr
- cindent

怪不得之前只设置noai和nosi不行，原来与这么多因素有关！ 但这样还是比较麻烦的，每次要粘贴的话，先set paste，然后粘贴，然后再set nopaste。有没有更方便的呢？你可能想到了，使用键盘映射呀，对。我们可以这样设置：：

```
:map <F10> :set paste<CR>
:map <F11> :set nopaste<CR>
```

这样在粘贴前按F10键启动paste模式，粘贴后按F11取消paste模式即可。其实，paste有一个

切换paste开关的选项，这就是pastetoggle。通过它可以绑定快捷键来激活/取消 paste模式。比如：：

```
:set pastetoggle=<F11>
```

这样减少了一个快捷键的占用，使用起来也更方便一些。 但，这是最方便的吗？Vimer们对高效的追求永无止境。还有其他更好地方法吗？ 你可能想到了，vim寄存器。对，使用vim寄存器 "+p 粘贴即可。根本不用考虑是否自动缩进，是否paste模式，直接原文传递！：

```
"+p
```

要说vim寄存器，就要从vim文件间的复制粘贴说起。 Vim中，若要复制当前行，普通模式下按 yy 即可，在要粘贴的地方按 p 。这是vim将复制内容保存到了自己的寄存器中的缘故。如果在其他地方执行yy，新的内容将覆盖掉原寄存器中内容。如果想保存原寄存器中内容而同时增加新的内容呢？这时就要在yy前增加标签了。标签以双引号开始，跟着的是标签名称，可以是数字0-9，也可以是26个字母，然后就是复制操作，这样就把复制内容保存到该标签寄存器里。通过下面命令显示所有寄存器内容：：

```
:reg
```

其中注意两个特殊的寄存器："* 和 "+。这两个寄存器是和系统相通的，前者关联系统选择缓冲区，后者关联系统剪切板。通过它们可以和其他程序进行数据交换。

备注：

> 若寄存器列表里无"* 或 "+ 寄存器，则可能是由于没有安装vim的图形界面所致。Debian/Ubuntu下可以通过安装vim-gnome解决。
>
> ```
> $ sudo apt-get install vim-gnome
> ```

选择缓冲区和系统剪切板啥子区别？让我们继续研究。

# 选择缓冲区和剪切板

不同于Windows，Linux系统里存在两个剪切板：一个叫做选择缓冲区(X11 selection buffer)，另一个才是剪切板(clipboard)。 选择缓冲区是实时的，当使用鼠标或键盘选择内容时，内容已经存在于选择缓冲区了，这或许就是选择缓冲区的由来吧。 使用下面的命令查看选择缓冲区的内容：：

```
$ xclip -out
```

如果没有xclip命令，Debian/Ubuntu下可以通过如下命令安装：：

```
$ sudo apt-get install xclip
```

可以使用鼠标中键或键入Shift+Insert来粘贴选择缓冲区的内容。但对于有些GUI程序，比如gedit，只能通过鼠标中键调用选择缓冲区的内容，使用Shift+Insert的话，调用的是剪切板的内容。 剪切板和Windows的剪切板类似，在选择文字内容后，执行Ctrl + c或在菜单里选择‘复制’的话，这时内容才存放到剪切板里。 使用下面的命令查看剪切板的内容：：

```
$ xclip -out -sel clipboard
```

而使用剪切板的内容，则是Ctrl+v。 但在有些情况下，比如gnome-terminal，不能直接使用Ctrl+c，Ctrl+v，这时就要用Shift+Ctrl+c，Shift+Ctrl+v代替。

# 原格式粘贴

好了，了解了选择缓冲区和剪切板，下面就是实现保留格式粘贴的完美解决方案：

- 方案一：

1. 选择文本内容
2. vim普通模式下按 ″*p 将选择缓冲区中内容粘贴进来

- 方案二：

1. 复制文件内容
2. vim普通模式下按 ″+ p 将剪切板内容粘贴进来

这时，如果要复制的内容也是vim编辑器中的内容，那么如何复制才更方便呢？

# vim中的复制

vim有一个可视模式(Visual Mode)，在此模式下可以选择区域。可以在普通模式下键入v进入可视模式，也可以个性化一点，键入V进入行可视模式，或者键入Ctrl+v进入列可视模式。这时移动光标就可以选择内容了。注意这时被选内容已经实时保存于选择缓冲区了，当然你也可以键入″+y将此内容也保存到剪切板里，或者″ay将内容保存到标签为a的寄存器中。但要知道，只有前两个中的内容可以在其他程序中使用，而a寄存器中的内容只能在该vim编辑器内使用。 也可以通过鼠标来复制。这里首先要打开鼠标模式。:

```
:set mouse=a
```

这样在普通模式下可以直接使用鼠标选择区域复制到选择缓冲区。但这种情况下不能复制到剪切板。 若要使用鼠标复制内容到剪切板，则需要做如下设置：:

```
:set mouse=v
```

这种情况下，除了可以像上面一样直接使用鼠标选择区域复制到选择缓冲区以外，还可以在右键菜单中选择“复制”来保存到剪切板里。但新问题又出来了。若显示行号，也会将行号一并选择。你会想到，这好办呀，如果不需要行号的话，在复制前，先执行set nonu来取消行号显示呗。 其实没必要这样，如果不需要复制行号的话，用在可视模式下用键盘来选择不就可以吗？ 并且，从上面的讨论，我们不难得出，使用选择缓冲区比使用剪切板要方便的多，可以节省很多步骤。 所以，最终我们得到了vim文件间复制粘贴的完美方案，文件传输的中转使用选择缓冲区。

# vim文件间复制粘贴完美方案

1. 在~/.vimrc中增加如下一行：:

   ```
   set mouse=v
   ```

2. 复制内容到选择缓冲区。
   - 带行号时，使用鼠标选择内容区域。
   - 不要行号，使用 ″*yny 复制n行或可视模式下选择。
3. 将选择缓冲区中内容粘贴到vim文件：普通模式下按 ″*p 。

---

补充： 设置vim中默认使用选择缓冲区寄存器 ″*：

```
set clipboard = unnamed
```

则可以直接通过y,p和系统选择缓冲区进行数据交换。

## Related Posts

| | | |
|---|---|---|
| 使用 git-svn 和 git-filter-branch 整理 SVN 版本库 | 2014-04-24 | 0 Comments |
| 复用 git.git 测试框架 | 2013-10-26 | 1 Comment |
| 墙不住的Git官网 | 2013-03-04 | 5 Comments |