

源程序文件（通常是纯文本文件）比较和合并工具一直是软件开发过程中比较重要的组成部分。现在市场上很多功能很强大的专用比较和合并工具，比如 **BeyondCompare**；很多 IDE 或者软件配置管理系统，比如 **Eclipse**, **Rational ClearCase** 都提供了内建的功能来支持文件的比较和合并。

当远程工作在 **Unix/Linux** 平台上的时候，恐怕最简单而且到处存在的就是命令行工具，比如 **diff**。可惜 **diff** 的功能有限，使用起来也不是很方便。作为命令行的比较工具，我们仍然希望能拥有简单明了的界面，可以使我们对比较结果一目了然；我们还希望能够在比较出来的多处差异之间快速定位，希望能够很容易的进行文件合并……。而 **Vim** 提供的 **diff** 模式，通常称作 **vimdiff**，就是这样一个能满足所有这些需求，甚至能够提供更多的强力工具。在最近的工作中，因为需要做很多的文件比较和合并的工作，因此对 **Vimdiff** 的使用做了一个简单的总结。我们先来看看 **vimdiff** 的基本使用。

启动方法

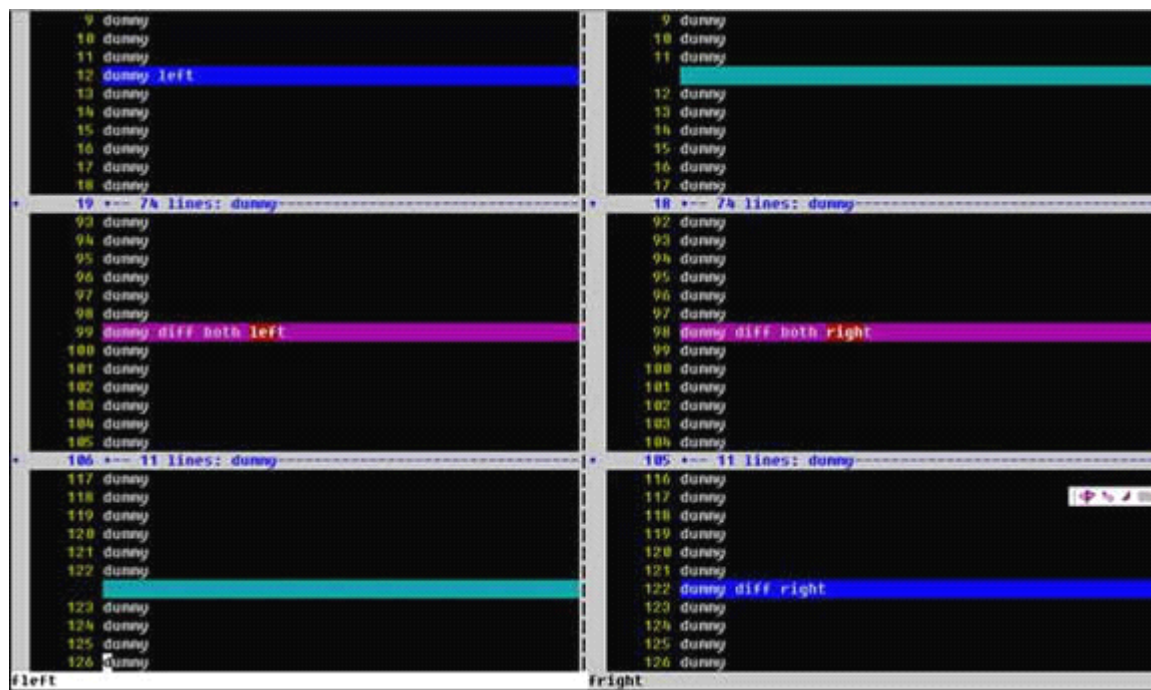
首先保证系统中的 **diff** 命令是可用的。**Vim** 的 **diff** 模式是依赖于 **diff** 命令的。**Vimdiff** 的基本用法就是：

```
# vimdiff FILE_LEFT FILE_RIGHT
```

或者

```
# vim -d FILE_LEFT FILE_RIGHT
```

图一就是 **vimdiff** 命令的执行结果的画面。



从上图我们可以看到一个清晰的比较结果。屏幕被垂直分割，左右两侧分别显示被比较的两个文件。两个文件中连续的相同的行被折叠了起来，以便使用者能把注意力集中在两个文件的差异上。只在某一文件中存在的行的背景色被设置为蓝色，而在另一文件中的对应位置被显示为绿色。两个文件中都存在，但是包含差异的行显示为粉色背景，引起差异的文字用红色背景加以突出。除了用这种方法启动 **vim** 的 **diff** 模式之外，我们还可以用分割窗口命令来启动 **diff** 模式：

```
# vim FILE_LEFT
```

然后在 **vim** 的 **ex** 模式（也就是"冒号"模式）下输入：

```
:vertical diffsplit FILE_RIGHT
```

也可以达到同样的效果。如果希望交换两个窗口的位置，或者希望改变窗口的分割方式，可以使用下列命令：

1. **Ctrl-w K**（把当前窗口移到最上边）
2. **Ctrl-w H**（把当前窗口移到最左边）
3. **Ctrl-w J**（把当前窗口移到最下边）
4. **Ctrl-w L**（把当前窗口移到最右边）

其中1和3两个操作会把窗口改成水平分割方式。

[回页首](#)

光标移动

接下来试试在行间移动光标，可以看到左右两侧的屏幕滚动是同步的。这是因为"**scrollbind**"选项被设置了的的结果，**vim** 会尽力保证两侧文件的对齐。如果不想要这个特性，可以设置：

```
:set noscrollbind
```

可以使用快捷键在各个差异点之间快速移动。跳转到下一个差异点：

```
]c
```

反向跳转是：

```
[c
```

如果在命令前加上数字的话，可以跳过一个或数个差异点，从而实现跳的更远。比如如果在位于第一个差异点的行输入"**2j**c"，将越过下一个差异点，跳转到第三个差异点。

[回页首](#)

文件合并

文件比较的最终目的之一就是合并，以消除差异。如果希望把一个差异点中当前文件的内容复制到另一个文件里，可以使用命令

```
dp (diff "put")
```

如果希望把另一个文件的内容复制到当前行中，可以使用命令

```
do (diff "get", 之所以不用 dg, 是因为 dg 已经被另一个命令占用了)
```

如果希望手工修改某一行，可以使用通常的 **vim** 操作。如果希望在两个文件之间来回跳转，可以用下列命令序列：

```
Ctrl-w, w
```

在修改一个或两个文件之后，**vimdiff** 会试图自动来重新比较文件，来实时反映比较结果。但是也会有处理失败的情况，这个时候需要手工来刷新比较结果：

```
:diffupdate
```

如果希望撤销修改，可以和平常用 **vim** 编辑一样，直接

```
<ESC>, u
```

但是要注意一定要将光标移动到需要撤销修改的文件窗口中。

[回页首](#)

同时操作两个文件

在比较和合并告一段落之后，可以用下列命令对两个文件同时进行操作。比如同时退出：

```
:qa (quit all)
```

如果希望保存全部文件：

```
:wa (write all)
```

或者是两者的合并命令，保存全部文件，然后退出：

```
:wqa (write, then quit all)
```

如果在退出的时候不希望保存任何操作的结果：

```
:qa! (force to quit all)
```

[回页首](#)

上下文的展开和查看

比较和合并文件的时候经常需要结合上下文来确定最终要采取的操作。**Vimdiff** 缺省是会把不同之处上下各 6 行的文本都显示出来以供参考。其他的相同的文本行被自动折叠。如果希望修改缺省的上下文行数，可以这样设置：

```
:set diffopt=context:3
```

可以用简单的折叠命令来临时展开被折叠的相同的文本行：

```
zo (folding open, 之所以用 z 这个字母，是因为它看上去比较像折叠着的纸)
```

然后可以用下列命令来重新折叠：

```
zc (folding close)
```

下图是设置上下文为3行，并展开了部分相同文本的 **vimdiff** 屏幕：

