

Linux - vim 的常用命令

作者: [douniwan](#)

原文链接: <https://ld246.com/article/1573047458423>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

 <https://b3logfile.com/bing/20171218.jpg?imageView2/1/w/960/h/540/interlace/1/q/100>

vim 的四种模式

正常模式(normal-mode)

插入模式(insert-mode)

命令模式(command-mode)

可视模式(visual-mode)

正常模式

用 vim 命令打开一个文件后，出来的那个界面就是正常模式的界面，在这个模式下你可以浏览文件的内容，也可以进行修改。然后不管处于什么界面中只要按 `ESC` 都会进入正常模式。处于该模式下的时候，输入不同的命令将产生不同的操作

移动

<blockquote>

<p>h: 当前光标向左移动</p>

<p>j: 当前光标向下移动</p>

<p>k: 当前光标向上移动</p>

<p>l: 当前光标向右移动</p>

<p>f{想要移动到的字符}: 移动到当前行的该光标后第一个该字符</p>

</blockquote>

复制粘贴剪切

<p>复制操作</p>

<blockquote>

<p>yy: 复制当前行</p>

<p>[n]yy: 复制当前行开始的 n 行</p>

<p>y\$: 复制当前光标到该行尾的东西</p>

</blockquote>

<p>粘贴操作</p>

<blockquote>

<p>p: 粘贴复制的内容</p>

</blockquote>

<p>剪切操作</p>

<blockquote>

<p>dd: 剪切当前行</p>

<p>[n]dd: 剪切当前行开始的 n 行</p>

</blockquote>

删除替换

<p>删除操作</p>

<blockquote>

<p>x: 删除当前光标指向的字符</p>

<p>d\$: 删除当前光标到行尾的内容</p>

<p>[n]x: 删除当前光标开始的 n 个字符</p>

<p>dd: 删除当前行（剪切会删除复制）</p>

<p>dw: 删除一个单词</p>

<p>J: 删除光标所在行与上一行之间的空格</p>

</blockquote>

<p>替换操作</p>

<blockquote>

<p>r: 替换当前光标的字符</p>

</blockquote>

<p>用法 想要被替换的字符 r 新字符</p>

撤销以及取消撤销

<blockquote>

<p>u: 撤销上次的操作, 如果有多次, 就按多次 u</p>

<p>ctrl + r: 取消撤销操作, 如果有多次, 就按多次</p>

</blockquote>

<h3 id="插入模式">插入模式</h3>

<p>在该模式下, 我们可以像在记事本中一样随意的插入删除文本。</p>

<p>在正常模式下不同的命令以不同的方式进入插入模式</p>

<blockquote>

<p>i: 在光标之前插入</p>

<p>a: 在光标之后插入</p>

<p>I: 在光标的行首插入</p>

<p>A: 在光标的行末插入</p>

<p>o: 在光标所在行的上面插入</p>

<p>O: 在光标所在行的下面插入</p>

<p>s: 删除光标后的一个字符, 然后进入插入模式</p>

<p>S: 删除光标所在的行, 然后进入插入模式</p>

</blockquote>

<h3 id="命令模式">命令模式</h3>

<h4 id="查找和高级替换">查找和高级替换</h4>

<blockquote>

<p>/要查找的内容: 如果有多个匹配则按 <code>n</code> 跳到下一匹配, 查找的内容可以是正表达式</p>

<p>:s/old/new: 用 new 替换当前行中第一次出现的 old</p>

<p>:s/old/new/g: 用 new 替换当前行中所有的 old</p>

<p>:n,m s/old/new/g: 用 new 替换从 n 到 m 行里面的所有 old</p>

<p>:%s/old/new/g: 用 new 替换当前文件里所有的 old</p>

</blockquote>

<h4 id="设置行号">设置行号</h4>

<blockquote>

<p>:set nu: 可以显示出行号, 只对当前 vim 有效, 如果想要每次 vim 后都有行号, 可以通过修改 v
m 的配置文件使得每次都生效相同的功能, 我的配置文件路径 <code>/etc/vim/vimrc</code></p>

</blockquote>

<h4 id="执行临时命令">执行临时命令</h4>

<blockquote>

<p>:! + 要执行的命令</p>

</blockquote>

<p>这个命令比较重要, 很多时候我们有可能需要查看一些东西, 但是又不愿意退出 vim 模式, 这
候就可以用这个命令。</p>

<h4 id="保存退出">保存退出</h4>

<blockquote>

<p>:w: 保存文件</p>

<p>:w!: 若文本属性为只读时, 强制保存</p>

<p>:q: 退出 vim</p>

<p>:q!: 不管编辑或未编辑都不保存退出</p>

<p>:wq: 保存, 退出</p>

<p>:wq!: 强制保存退出</p>

<p>:e!: 将文档还原成最原始状态</p>

<p>:x: 同样也是保存退出, 类似 <code>:wq</code></p>

</blockquote>

<h3 id="可视模式">可视模式</h3>

<h4 id="什么是可视模式-">什么是可视模式? </h4>

<p>可视模式简单来说就是选中一块编辑区域, 然后在上面执行一些操作, 比如删除, 替换, 改变大
写等。</p>

<h4 id="选中方式">选中方式</h4>

v 进入字符可视化模式
V 进入行可视化模式
Ctrl+v 进入块可视化模式

<h4 id="选中光标到行尾的内容">选中光标到行尾的内容</h4>
<blockquote>
<p>v + f{光标的内容} + \$</p>
</blockquote>
<h4 id="选中里面的内容">选中 <code>" "</code> 里面的内容</h4>
<blockquote>
<p>f + 首字母 + vi + "</p>
</blockquote>
<h4 id="再次选择上次选中的区域">再次选择上次选中的区域</h4>
<p>（没有被删除的情况下）可以使用</p>
<blockquote>
<p>gv</p>
</blockquote>
<h4 id="切换当前光标的端点">切换当前光标的端点</h4>
<p>选中的区域是以两个端点（左上，右下）来标识的，有时候需要在两个端点之间切换，就可以使用该命令</p>
<blockquote>
<p>o</p>
</blockquote>
<h4 id="其他操作">其他操作</h4>
<p>在选中的区域上也可以使用正常模式下的一些命令，以及一些其他命令</p>
<blockquote>
<p>d: 删除选中文本

c: 修改选中文本

r: 替换选中文本

I: 在选中文本前插入

A: 在选中文本后插入

gu: 选中区域转为小写

gU: 选中区域转为大写

g~: 大小写互调</p>
</blockquote>
<p>大体上 vim 中所接触到的命令就如此了，后面接触到再继续补充</p>