

1. Linux实操篇- Vi和Vim编辑器

1.1 vi和vim的基本介绍

在Linux系统中，vi文本编辑器是系统内置的重要工具。它功能基础且实用，是Linux环境下进行文本编辑的常用方式。

而Vim则是在vi基础上的强化升级版本。其最大的亮点在于具备强大的程序编辑能力，能够主动通过字体颜色来辨别语法的正确性，这一特性极大地方便了程序员进行代码编写，能快速识别代码中的潜在问题。除此之外，Vim还拥有极为丰富的辅助编程功能，如代码补完功能，可根据输入的代码片段自动补全可能的代码内容，提高编程效率；编译功能允许在编辑器内直接对代码进行编译操作，无需切换到外部编译环境；错误跳转功能则能在编译出现错误时，快速定位到错误代码行，节省排查问题的时间。这些功能使得Vim在程序员群体中得到了广泛的应用与青睐。

6.2 vi 和 vim 常用的三种模式

6.2.1 正常模式

当使用vim打开一个文件时，默认进入的就是正常模式。在此模式下，用户可以通过“上下左右”方向键来灵活移动光标，精准定位到需要编辑的位置。对于文件内容的处理，能够使用“删除字符”（如按“x”键删除光标所在位置的单个字符）或“删除整行”（按“dd”键删除光标所在的整行）操作来清理不需要的内容。同时，“复制”（如按“yy”键复制光标所在行）和“粘贴”（按“p”键将复制的内容粘贴到光标之后）功能也可以高效地处理文件数据，实现内容的快速复用与调整。

6.2.2 插入模式

若要对文件内容进行编辑，需要从正常模式切换到插入模式。按下“i”“I”“o”“O”“a”“A”“r”“R”等字母中的任意一个均可进入编辑模式，其中最为常用的是按下“i”键，它会在光标当前位置之前插入文本，方便用户直接输入新的内容。

6.2.3 命令行模式

从正常模式切换到命令行模式，需要先按下“esc”键确保处于正常模式状态，再输入“:”。在命令行模式中，用户可以执行一系列的指令操作。例如，“:w”用于保存文件，“:q”用于退出vim，“:wq”则表示保存并退出；“:s/old/new/g”可以实现对文件中指定内容的替换，将“old”替换为“new”，“g”表示全局替换；“:set nu”用于显示文件的行号，“:set nonu”则用于取消行号显示。通过这些指令，能够完成读取、存盘、替换、离开vim以及显示行号等一系列重要的操作。

6.3 vi 和 vim 基本使用

以使用vim开发一个名为“Hello.java”的程序为例，详细步骤如下：

1. **打开编辑器**：在Linux系统的终端中输入“vim Hello.java”，按下回车键后，即可启动vim编辑器并创建一个名为“Hello.java”的文件。
2. **进入插入模式**：按下“i”键，进入插入模式，此时可以在文件中输入Java代码。例如：

```
java
```

```
public class Hello {  
    public static void main(String[] args) {  
        System.out.println("Hello, world!");  
    }  
}
```

1. **保存并退出**：输入完成后，按下“esc”键退出插入模式，回到正常模式。接着输入“:wq”，按下回车键，即可保存文件内容并退出 vim 编辑器。

6.4 各种模式的相互切换

1. **正常模式→插入模式**：按下“i”“I”“o”“O”“a”“A”“r”“R”等键，其中“i”在光标前插入，“I”在当前行首插入，“o”在光标下新开一行插入，“O”在光标上新开一行插入，“a”在光标后插入，“A”在当前行尾插入，“r”替换光标所在字符，“R”进入替换模式。
2. **插入模式→正常模式**：按下“esc”键，即可从插入模式切换回正常模式，以便进行光标移动、复制、删除等操作。
3. **正常模式→命令行模式**：按下“esc”键确保处于正常模式，再输入“:”，即可进入命令行模式，执行保存、退出、替换等指令。
4. **命令行模式→正常模式**：在命令行模式下执行完指令后，按下“esc”键，即可回到正常模式，继续对文件进行常规操作。

6.5 vi 和 vim 快捷键

6.5.1 快捷键使用练习

1. 拷贝与粘贴
 - **拷贝当前行**：在正常模式下，将光标移动到需要拷贝的行，按下“yy”键，即可将当前行内容复制到缓冲区。
 - **拷贝当前行向下的 5 行**：将光标移动到起始行，按下“5yy”键，即可将从当前行开始向下的 5 行内容复制到缓冲区。
 - **粘贴**：完成拷贝后，将光标移动到需要粘贴的位置，按下“p”键，缓冲区中的内容将被粘贴到光标之后。
2. 删除操作
 - **删除当前行**：在正常模式下，将光标移动到需要删除的行，按下“dd”键，该行内容将被删除。
 - **删除当前行向下的 5 行**：将光标移动到起始行，按下“5dd”键，从当前行开始向下的 5 行内容将被删除。
3. **查找操作**：在命令行模式下，输入“/关键字”，然后按下回车键，vim 将从文件当前位置开始向下查找指定的“关键字”。若要查找下一个匹配项，可在查找到第一个匹配项后，按下“n”键；若要查找上一个匹配项，可按下“N”键。
4. 行号设置
 - **显示行号**：在命令行模式下，输入“:set nu”，按下回车键，文件将显示行号，方便定位和引用特定行。
 - **取消行号**：在命令行模式下，输入“:set nonu”，按下回车键，文件的行号将被取消显示。
5. 快速定位
 - **到文档最末行**：编辑“/etc/profile”文件时，在正常模式下，按下“G”键，光标将快速移动到文档的最后一行。
 - **到文档最首行**：在正常模式下，按下“gg”键，光标将快速移动到文档的第一行。

6. **撤销操作**：在一个文件中输入“hello”后，若处于正常模式，按下“u”键，即可撤销上一次的输入操作，“hello”将被删除。
7. **指定行定位**：编辑“/etc/profile”文件时，在正常模式下，将光标移动到合适位置，输入数字“20”，再按下“shift+g”键，光标将移动到第 20 行。

除上述快捷键外，还有更多丰富的快捷键可供使用，详细内容可查看整理的文档资料，以进一步提升在 vi 和 vim 编辑器中的操作效率。同时，为了更直观地记忆快捷键，可参考快捷键的键盘对应图，熟悉不同按键组合在 vi 和 vim 中的功能。

6.6 vi 和 vim 课堂练习

1. **编写 Hello.java 程序**：使用 vim 或 vi 编写一个完整的“Hello.java”程序，按照 6.3 节中的步骤，从创建文件、输入代码到保存退出，熟悉整个流程。
2. **快捷键练习**：将课堂上老师讲解的各种快捷键亲自操作一遍，通过实际操作加深对快捷键的理解和记忆，包括但不限于拷贝、粘贴、删除、查找、定位等操作的快捷键，确保能够熟练运用这些快捷键提高文本编辑和程序开发的效率。

2. 课后练习

一、选择题

1. 以下关于 vi 和 Vim 的说法，正确的是（ ）
 - A. vi 是 Vim 的增强版本
 - B. Vim 不能辨别语法正确性
 - C. vi 系统不会内置，需要单独安装
 - D. Vim 具有代码补完、编译等丰富功能
2. 当使用 vim 打开文件后，默认进入的模式是（ ）
 - A. 插入模式
 - B. 命令行模式
 - C. 正常模式
 - D. 编辑模式
3. 在 vim 的命令行模式中，用于保存并退出的指令是（ ）
 - A. :w
 - B. :q
 - C. :wq
 - D. :set nu
4. 在 vim 正常模式下，想要删除光标所在行向下的 3 行，应使用的快捷键是（ ）
 - A. 3dd
 - B. dd3
 - C. 3yy
 - D. yy3

二、简答题

1. 请简述在 vim 中如何从插入模式切换到命令行模式。
2. 对比 vim 中“i”“I”“o”“O”这四个按键进入插入模式的区别。

三、实操题

1. 假设你需要在 Linux 系统中使用 vim 编写一个简单的 C++ 程序 “test.cpp”，实现输出 “Hello, C++! ”。请写出详细步骤，包括模式切换、代码输入、保存退出等操作。
2. 现有一个文本文件 “example.txt”，请使用 vim 进行以下操作：
 - 在文件中查找单词 “apple”，并定位到下一个出现的位置。
 - 显示文件的行号。
 - 将文件中所有的 “apple” 替换为 “banana”。
 - 最后保存并退出。