

Building Your Own Vim-like Text Editor

Contributor: 张泽凯 12442026@mail.sustech.edu.cn

今天，我们将动手打造一个迷你版的 Vim。Vim 是一个强大的文本编辑器，其可以追溯到 20 世纪 70 年代的 Vi，1991 年演变为 Vim（意为“Vi IMproved”）。这款工具以纯键盘操作为核心，让用户能够精准高效地掌控代码和文本。尽管它以学习曲线陡峭著称，一旦你熟悉了它，你会发现它不仅提升了文本处理效率和控制力，还让操作过程看起来非常炫酷。

Welcome, future Vim-masters! Today, you begin your journey toward creating a mini version of Vim, the mighty text editor loved by programmers across the globe. Originally birthed in the '70s as **Vi**, and then leveled up in 1991 as **Vim** ("Vi IMproved"), Vim lets users dive deep into their code with pure keyboard power. Though it has a reputation for being challenging to learn, many find that mastering Vi/Vim significantly boosts productivity and control when working with text.



Project Overview

这个项目有基础要求和进阶要求两个层级。基本要求实现了 MiniVim 的核心功能，而高级要求会实现更加炫酷的功能，为那些希望实现更多功能的同学提供挑战。

The project is divided into two levels. The basic level satisfies the essential requirements for functionality, while the advanced level is for those who want to push the boundaries and implement additional features.

Basic Requirements

Vim 有三个主要的模式：

1. **普通模式**：用于移动光标和执行操作（如删除、复制）。
2. **插入模式**：用于输入文本内容。
3. **命令模式**：用于执行保存、退出等命令。

Vim has three primary modes: **Normal**, **Insert**, and **Command-Line**. Each mode serves a specific purpose.

1. **Normal Mode** - Navigate, edit, and more.
2. **Insert Mode** - Type and modify text.
3. **Command-Line Mode** - Save, search, or quit with commands like `:w`, `:q`, and more.

Normal Mode

普通模式是您打开 Vim 时的默认模式。在普通模式下，你并不能直接输入文本，而是移动光标，执行删除、复制、粘贴等操作。通过普通模式，你可以用最少的按键快速完成一些复杂的文本操作。你可以将其看作 Vim 的“控制中心”。

Normal Mode is the **default** mode when you open Vim. In Normal mode, you're not directly typing text but instead issuing commands to navigate, delete, copy, paste, and manipulate text. This mode allows you to quickly perform complex text operations with minimal keystrokes.

Think of it as the "control center".

- **File and Mode Commands:**

- `:w` - 保存文件. Save the file.
- `:q` - 退出编辑器. Quit the editor.
- `:wq` - 保存并退出编辑器. Save and quit.
- `i` - 切换到插入模式. Switch to Insert mode.
- 按下`:`进入命令模式以执行其他命令. Press `:` to open **Command-Line mode** for more advanced commands.

- **Cursor Movement:**

- 用 `h`, `j`, `k`, `l` 控制光标左、下、上、右移动. Move with `h` (left), `j` (down), `k` (up), and `l` (right).
- 用方向键控制光标移动. Arrow keys also work for cursor navigation.
- **12/10 UPDATE:** 当文本内容太长时，可以查看溢出屏幕的内容（比如用方向键）。When the content exceeds the visible area, scrolling should be enabled to ensure all text remains accessible.

- **Line Navigation:**

- `0` - 光标移动到当前行的行首. Jump to the beginning of the current line.
- `$` - 光标移动到当前行的行尾. Jump to the end of the current line.
- `gg` - 光标移动到第一行的行首. Go to the first line with the cursor at the beginning.
- `G` - 光标移动到最后一行的行首. Go to the last line with the cursor at the beginning.

- **Basic Editing:**

- `dd` - 删除当前行. Delete the current line.
- `yy` - 复制当前行. Copy the line.
- `p` - 粘贴复制的内容. Paste the copied line below the cursor.

Insert Mode

插入模式类似于传统的编辑器，在该模式下，你输入的所有内容都会直接显示在屏幕上。在普通模式下按下 `i` 可以进入插入模式，按下 `Esc` 可以返回普通模式。

Insert Mode allows direct text input. Here, everything you type appears on the screen (like a traditional editor). As mentioned above, activate it by pressing `i` in Normal Mode, and return to Normal Mode using `Esc`.

- **Direct Text Insertion:** 在当前光标处插入字符. Type characters that display at the cursor position.
- **New Line Creation:** 按下 `Enter` 会创建新的一行，并且光标会移动到新行的行首. `Enter` inserts a new line, moving the cursor to the beginning of the new line.
- **Backspace Support:** 按下 `Backspace` 删除字符，光标随之移动（注意删除一行时光标的移动. Delete characters with `Backspace`. Navigate up to delete across lines if at the cursor is at the start of a line.
- **Real-Time Display Update:** 文本的改动应该实时反映的屏幕上. Changes should instantly reflect in the text.
- **Cursor Movement with Typed Text:** 注意文本编辑时光标的移动。12/10 UPDATE: 此外，需要像普通模式一样可以用方向键移动光标和查看溢出内容. Cursor should keep up with typing, mimicking typical editor behavior. When the content exceeds the visible area, scrolling should be enabled to ensure all text remains accessible.
- **Exit to Normal Mode:** 按下 `Esc` 返回普通模式. `Esc` should switch back to Normal mode.

: Command-Line Mode

命令模式用于文件操作、搜索等功能。要从普通模式进入命令模式，只需按下 `:`，然后输入命令，例如 `:w`（保存）、`:q`（退出）或 `:wq`（保存并退出），并按下 `Enter` 执行。此处不再罗列命令模式的基本要求，因为命令模式在文档前面已作为普通模式功能的一部份实现。

Command-Line mode is used for file operations, search, and more. To enter Command-Line mode from Normal mode, simply press `:`, then complete commands such as `:w` (save), `:q` (quit), or `:wq` (save and quit), and press `Enter` to execute them. Note that Command-Line mode has been implemented as part of the requirements in Normal mode.

Advanced Features

如果你完成了基本功能，并希望挑战更加高级的功能和获得更高的分数，以下是高级功能的建议（不局限于以下功能）：

Here are some ways to make your mini-Vim truly powerful! Below are some possible features you can work on, and you may **customize them as you wish**, as long as they make sense for the project.

- **Search and Replace:** 实现搜索和替换功能，比如 `:s/old/new/g` 可以将文档中的所有的 `old` 替换为 `new`. Implement find and replace with a command like `:s/old/new/g`.
- **Multi-File Editing:** 可以同时打开文件，并且在不同文件之前切换进行编辑. Open and switch between multiple files seamlessly.

- **Undo and Redo:** 实现撤销和恢复功能. Let users undo or redo their last action.
- **Line Numbers and Jumping:** 显示行号并且可以跳转到指定行号. Show line numbers and jump to a specific line.
- **UI Improvements:** 把编辑器界面变得更加美观! Add some pizzazz! Use colors or styles to make different modes or commands stand out.

Submission Requirements

- **Source Code:** 项目相关的所有源文件. Submit all relevant source files for the project.
- **Project Documentation:** 一个简洁明了的文档, 里面包括你实现的功能的用途和用法. Write a **brief** doc that explains your design choices, features, and usage instructions.

Grading Criteria

Category	Criteria	Points
Basic Requirements (80 points)		
Normal Mode & Command-Line Mode (40 points)	File and Mode Commands	20
	- :w Save file	4
	- :q Quit editor	4
	- :wq Save and quit	4
	- i Switch to Insert mode	4
	- : Open Command-Line mode	4
	Cursor Movement	10
	- Move with h/j/k/l	2
	- Arrow key navigation	2
	- 0 and \$ for line start and end	2
	- gg and G for first and last line navigation	2
	- View overflow content	2
	Line Operations	10
	- dd Delete current line	4
	- yy Copy current line and p Paste copied line	6
Insert Mode (25 points)	Text Insertion	20
	- Type characters at cursor position	4
	- Enter capabilities	4
	- Backspace capabilities	4
	- Real-time cursor updates during typing	4
	- View overflow content	4
	Mode Switching	5
	- Press Esc to return to Normal mode	5

Project Completeness and Documentation (15 points)

Source Code Completeness	5
- All files compile and run correctly	5
Documentation	10
- Brief and clear design and usage explanation	10

Advanced Features (Maximum 40 points)

Search and Replace	10.
- Implement <code>:s/old/new/g</code> command	10 Max.
Multi-File Editing	20
- Support for opening and switching between files	20 Max.
Undo and Redo	10
- Implement <code>u</code> (undo) and <code>Ctrl+R</code> (redo)	10 Max.
Line Numbers and Jumping	10
- Show line numbers and jump to specific lines	10 Max.
...	

Hint

如果你对某些行为不确定（例如某些情况下的光标移动），可以先看看 Vim 实际是如何处理的。

如果你考虑使用 `<ncurses.h>` 来实现基于终端的功能，以下是一些推荐的学习资料：

- NCURSES Programming HOWTO: <https://tldp.org/HOWTO/NCURSES-Programming-HOWTO/>
- NCURSES Programming HOWTO 中文翻译: <https://poet.cosoft.org.cn/downloads/docs/ncurses-cn-2nd.pdf>

如果您不熟悉 Vi/Vim，可以先尝试下：

- Vim交互式教程: [Open Vim](#)

If you are not sure about certain behaviors (e.g. cursor movement in some cases), you may check out what vim actually does.

You might consider using `<ncurses.h>` for terminal-based capabilities. Here are some good resources to start with:

- NCURSES Programming HOWTO: <https://tldp.org/HOWTO/NCURSES-Programming-HOWTO/>
- NCURSES Programming HOWTO 中文翻译: <https://poet.cosoft.org.cn/downloads/docs/ncurses-cn-2nd.pdf>

If you are not familiar with Vi/Vim, try

- an interactive tutorial: <https://openvim.com/>
-

Good luck! After this project, you might be the one laughing at all those "How to Exit Vim" memes. 😄