Vim

Sumner Evans

September 7, 2017

Mines Linux Users Group

Text Editors

Why do we need text editors when we have IDEs?

- Often need to edit configuration files and you don't want to fire up Visual Studio to edit a simple text file.
- Most tasks do not need any fancy IDE features to complete. For example, writing a simple Markdown file, writing a LaTeX document, writing this presentation with Beamer, programming, etc.
- Most of the time, users don't use nearly all of the features that IDEs provide. This is a waste of computer resources.
- Good text editors have the ability to be extended to bring IDE-like features to your text editor. This gives greater customisation ability to the user.

Why do we need text editors when we have IDEs?

- Often need to edit configuration files and you don't want to fire up Visual Studio to edit a simple text file.
- Most tasks do not need any fancy IDE features to complete. For example, writing a simple Markdown file, writing a LaTeX document, writing this presentation with Beamer, programming, etc.
- Most of the time, users don't use nearly all of the features that IDEs provide. This is a waste of computer resources.
- Good text editors have the ability to be extended to bring IDE-like features to your text editor. This gives greater customisation ability to the user.

Why do we need text editors when we have IDEs?

- Often need to edit configuration files and you don't want to fire up Visual Studio to edit a simple text file.
- Most tasks do not need any fancy IDE features to complete. For example, writing a simple Markdown file, writing a LaTeX document, writing this presentation with Beamer, programming, etc.
- Most of the time, users don't use nearly all of the features that IDEs provide. This is a waste of computer resources.
- Good text editors have the ability to be extended to bring IDE-like features to your text editor. This gives greater customisation ability to the user.

Why do we need text editors when we have IDEs?

- Often need to edit configuration files and you don't want to fire up Visual Studio to edit a simple text file.
- Most tasks do not need any fancy IDE features to complete. For example, writing a simple Markdown file, writing a LaTeX document, writing this presentation with Beamer, programming, etc.
- Most of the time, users don't use nearly all of the features that IDEs provide. This is a waste of computer resources.
- Good text editors have the ability to be extended to bring IDE-like features to your text editor. This gives greater customisation ability to the user.

Why use a terminal-based text editor?

- Often need to edit files and you don't want to have to open a GUI file selector to navigate to the correct file and edit it. (In fact, there are some bad file selectors which don't even let you open dotfiles!)
- If you are SSH-d into a remote machine (for example, while administering a server), you may not have the luxury of a GUI (there is SSH tunnelling, but that is not always an option).
- These text editors are made by programmers for programmers.

Why use a terminal-based text editor?

- Often need to edit files and you don't want to have to open a GUI file selector to navigate to the correct file and edit it. (In fact, there are some bad file selectors which don't even let you open dotfiles!)
- If you are SSH-d into a remote machine (for example, while administering a server), you may not have the luxury of a GUI (there is SSH tunnelling, but that is not always an option).
- These text editors are made by programmers for programmers.

Why use a terminal-based text editor?

- Often need to edit files and you don't want to have to open a GUI file selector to navigate to the correct file and edit it. (In fact, there are some bad file selectors which don't even let you open dotfiles!)
- If you are SSH-d into a remote machine (for example, while administering a server), you may not have the luxury of a GUI (there is SSH tunnelling, but that is not always an option).
- These text editors are made by programmers for programmers.

Vim

Why use Vim over other terminal-based text editors?

- You don't need a mouse at all. In fact, by default, you *can't* use your mouse.
- You can do everything with the keyboard in just a few keystrokes.
- The `CTRL` key is hard to reach on standard keyboards.
- But all of this is only possible because of **modal editing**.

Why use Vim over other terminal-based text editors?

- You don't need a mouse at all. In fact, by default, you *can't* use your mouse.
- You can do everything with the keyboard in just a few keystrokes.
- The `CTRL` key is hard to reach on standard keyboards.
- But all of this is only possible because of **modal editing**.

Why use Vim over other terminal-based text editors?

- You don't need a mouse at all. In fact, by default, you *can't* use your mouse.
- You can do everything with the keyboard in just a few keystrokes.
- The `CTRL` key is hard to reach on standard keyboards.
- But all of this is only possible because of **modal editing**.

Why use Vim over other terminal-based text editors?

- You don't need a mouse at all. In fact, by default, you *can't* use your mouse.
- You can do everything with the keyboard in just a few keystrokes.
- The `CTRL` key is hard to reach on standard keyboards.
- But all of this is only possible because of **modal editing**.

Non Modal Editing

- In most editors, when you type with your keyboard, what you type is inserted right at the cursor location.
- To access functionality such as selection and scrolling you need to use your mouse.
- To access functionality such as find [and replace], you need to use a menu item or some complicated keyboard shortcut.

Non Modal Editing

- In most editors, when you type with your keyboard, what you type is inserted right at the cursor location.
- To access functionality such as selection and scrolling you need to use your mouse.
- To access functionality such as find [and replace], you need to use a menu item or some complicated keyboard shortcut.

Non Modal Editing

- In most editors, when you type with your keyboard, what you type is inserted right at the cursor location.
- To access functionality such as selection and scrolling you need to use your mouse.
- To access functionality such as find [and replace], you need to use a menu item or some complicated keyboard shortcut.

Modal Editing: The Ultimate Separation of Powers

- Pressing the same keys do different actions depending on which mode you are in.
- This is extremely space-efficient.

Modes: Normal

- This is the default (normal) mode.
- It is sometimes referred to as command mode.
- This is the mode that you are in when you enter Vim.
- You can tell you are in normal mode because there will be no text in the bottom left corner of your console window.
- Used to get to other modes, for cursor movement, copy/pasting, saving, etc. . .
- **To return here from other modes, press the `ESC` key.**

Modes: Insert

- Allows you to actually type text.
- To get to insert mode from normal mode, press `i`

Modes: Visual

- Allows you to select text and perform actions upon that selected text.

Installing Vim

- Linux: install the `vim` package using your package manager
- macOS: install the `vim` using Homebrew
- Windows: Google it

Further Reading

- My configurations:
  `https://github.com/sumnerevans/dotfiles`

References

I used Caleb Jhones' and Jack Rosenthal's presentations on Vim from last year as as inspiration (maybe bordering on plagiarism) for this talk.

- Caleb's presentation: `https://github.com/ThirdOf5/VIM-Intro`
- Jack's presentation: `https://github.com/jackrosenthal/lug-vim-awesome`

Questions?

Copyright Notice

Colorado School of Mines
Linux Users Group