

8 Text Editors

8.1 Installation and Start & Quit

8.1.1 安装 emacs

8.1.2 启动 emacs

8.1.3 Short-cut 保存与关闭 emacs

8.2 Customize Key binding 自定义按键

8.2.1 Basic: 编辑 init 文件

8.2.2 重置所有configuration

8.2.3 直接在 Emacs 中创建文件

8.3 Pro-tips

8.3.1 Auto-Complete

8.3.2 Company Mode

8.3.3 添加 Emacs Shortcut term

8.3.4 Editing remotely with TRAMP

8.3.5 Text-only mode

8 Text Editors

[Emacs](#) is an extensible, customizable, free, text editor. 它有一个 IDE mode 能够使用 `gdb` and `lldb`. 掌握 Emacs 可以让你 edit code *very* quickly.

Reasons to learn Emacs:

- Endlessly customizable.
- key bindings 快速编辑代码.
- 具有 IDE 特性还 with zero project setup.
- 可选择 text-only mod, 适合 remote servers
- **Emacs keyboard shortcuts work in many places: command line, GDB prompt, LLDB prompt, Xcode (optional), Visual Studio (optional)**

8.1 Installation and Start & Quit

8.1.1 安装 emacs

安装

```
brew install --cask emacs # macOS
sudo apt install emacs    # Windows/WSL, Linux
```

确认 compiler 版本

```
g++ --version # macOS
Apple clang version 13.1.6 (clang-1316.0.21.2.5)
$ lldb --version # macOS
Apple Swift version 5.6.1 (swiftlang-5.6.0.323.66 clang-1316.0.20.12)
$ g++ --version # WSL/Linux
g++ (GCC) 8.5.0 20210514
$ gdb --version # WSL/Linux
GNU gdb (GDB)
```

8.1.2 启动 emacs

启动 emacs:

```
emacs
```

在后台启动 Emacs，以便继续使用terminal:

```
emacs &
```

8.1.3 Short-cut 保存与关闭 emacs

保存: `C-x C-s`

`C` 表示 control 键. 这里的意思是按住 `C`，而后按住 `x` 而松开，而后按住 `s` 而松开，最后松开 `C`。

关闭: `C-x C-c`

按住 `C`，而后按住 `x` 而松开，而后按住 `c` 而松开，最后松开 `C`。

8.2 Customize Key binding 自定义按键

8.2.1 Basic: 编辑 init 文件

通过下面这个指令可以打开 emacs 的 init 文件.

```
emacs ~/.emacs.d/init.el
```

然后开始编辑 init.el 文件.

注意这个文本格式中 `;` 就是注释的意思.

1. 对于 Mac 而言，将 Command 映射到 Meta，将 Option 映射到 Super 更符合人体工程学.

```
;; macOS modifier keys
(setq mac-command-modifier 'meta) ; Command == Meta
(setq mac-option-modifier 'super) ; Option == Super
```

2. basic customization

```
;; Don't show a startup message
(setq inhibit-startup-message t)

;; Show line and column numbers
(setq line-number-mode t)
(setq column-number-mode t)

;; Show syntax highlighting
(global-font-lock-mode t)

;; Highlight marked regions
(setq-default transient-mark-mode t)

;; Parentheses
(electric-pair-mode 1)                ; automatically close parentheses, etc.
(show-paren-mode t)                  ; show matching parentheses

;; Smooth scrolling (one line at a time)
(setq scroll-step 1)

;; Tab settings: 2 spaces. See also: language-specific customizations below.
(setq-default indent-tabs-mode nil)
(setq tab-width 2)

;; Easier buffer switching
(global-set-key "\C-x\C-b" 'electric-buffer-list)
```

3. package manager

automates package installation and configuration.

```
;; Configure built in package manager
(require 'package)
(package-initialize)
(add-to-list 'package-archives '("melpa" . "https://melpa.org/packages/"))
(add-to-list 'package-archives '("gnu" . "https://elpa.gnu.org/packages/"))

;; Install use-package
;; https://github.com/jwigley/use-package
(when (not (package-installed-p 'use-package))
  (package-refresh-contents)
  (package-install 'use-package))
(eval-when-compile
  (require 'use-package))
```

4. undo/redo

```
;; More intuitive undo/redo. M-_ undo, C-M-_ redo
(use-package undo-tree
  :config
  (global-undo-tree-mode)
  (global-set-key "\C-\M-_" 'redo)
  :ensure t
  :defer t
)
```

5. C/C++

```
;; Intellisense syntax checking
;; http://www.flycheck.org/en/latest/
(use-package flycheck
  :config
  ;; enable in all modes
  (global-flycheck-mode)
  ;; C++17
  (add-hook 'c++-mode-hook (lambda () (setq flycheck-clang-language-standard "c++17"))))
  :ensure t
  :defer t
)

;; C and C++ programming. Build with C-c m. Rebuild with C-c c. Put
;; this in c-mode-base-map because c-mode-map, c++-mode-map, and so
;; on, inherit from it.
(add-hook 'c-initialization-hook
  (lambda () (define-key c-mode-base-map (kbd "C-c m") 'compile)))
(add-hook 'c-initialization-hook
  (lambda () (define-key c-mode-base-map (kbd "C-c c") 'recompile)))
(setq-default c-basic-offset tab-width) ; indentation
(add-to-list 'auto-mode-alist '("\\.h$" . c++-mode)) ; assume C++ for .h files
```

6. integrated debugging

如果是 wsl 则把这里的 lldb 换成 gdb

```
(use-package realgud
  :ensure t
)
(use-package realgud-lldb
  :ensure t
)
```

(使用debug: `M-x gud-gdb` for WSL/Linux, `M-x realgud--lldb` for mac)

8.2.2 重置所有configuration

重开命令：

```
rm -rf ~/.emacs ~/.emacs.d/
```

8.2.3 直接在 Emacs 中创建文件

```
emacs 1.cpp
```

8.3 Pro-tips

8.3.1 Auto-Complete

`M + /` 可以提供 C++ 语句的 auto complete.

8.3.2 Company Mode

添加这个到 init.el

```
;; Autocomplete for code
;; Company docs: https://company-mode.github.io/
;; Company TNG: https://github.com/company-mode/company-mode/issues/526
(use-package company
  :config
  (company-tng-configure-default)      ; use default configuration
  (global-company-mode)
  :ensure t
  :defer t                             ; lazy loading
)
```

8.3.3 添加 Emacs Shortcut term

```
e ()
{
  emacs "$@" &
}
```

添加这个到 `~/.bash_profile` / `~/.zshrc`，我们就可以直接用一个 `e` 来指代 `emacs`

```
e stats.cpp
```

8.3.4 Editing remotely with TRAMP

Emacs [TRAMP](#) mode lets you edit a file on a remote server using a local GUI window.

Configure Emacs TRAMP mode to use SSH multiplexing. Add this to your `~/.emacs.d/init.el`.

```
(use-package tramp
  :config
  (setq tramp-default-method "ssh")
  (setq tramp-ssh-controlmaster-options
    (concat
      "-o ControlMaster auto "
      "-o ControlPath ~/.ssh/socket-%%C "
    ))
  (setq tramp-use-ssh-controlmaster-options nil)
  :defer 1 ; lazy loading
)
```

SSH into your remote server, CAEN Linux in this example. This will set up an SSH multiplexing connection.

```
ssh login.engin.umich.edu
...
```

In Emacs, open the file `/ssh:login.engin.umich.edu:main.cpp`. Recall `C-x C-f` is `find-file`. Tab completion works in the minibuffer. You're now editing a file `main.cpp` on a remote server.

```
ssh login.engin.umich.edu
hostname
# ...caen-vnc-vm05.engin.umich.edu
cd pl-stats
tmux new -s shared
emacs -nw main.cpp
```

Bob connects to the same remote server that Alice did. He connects to Alice's tmux session named `shared`. She's already running Emacs, so he sees her Emacs session.

```
ssh caen-vnc-vm05.engin.umich.edu
tmux attach -t shared
```

8.3.5 Text-only mode

If you're on a remote server without a GUI, you can use Emacs in text-only mode. The `-nw` option stands for "no window".

```
emacs -nw
```

