

Emacs 入门

@yinhm

GZLUG GZTechParty

March 20, 2011

**GNU Emacs is an extensible,
customizable text editor —and
more.**

介绍

一点历史

介绍

一点历史

- 1970 年, (Editing MACroS) 诞生于 MIT 人工智能实验室

介绍

一点历史

- 1970 年, (Editing MACroS) 诞生于 MIT 人工智能实验室
- 1984 年, Stallman 开发 GNU Emacs 用来代替 Gosling Emacs

介绍

一点历史

- 1970 年, (Editing MACroS) 诞生于 MIT 人工智能实验室
- 1984 年, Stallman 开发 GNU Emacs 用来代替 Gosling Emacs
- 1985 年 3 月 20 日, Version 13 作为第一个 public release

介绍

一点历史

- 1970 年, (Editing MACroS) 诞生于 MIT 人工智能实验室
- 1984 年, Stallman 开发 GNU Emacs 用来代替 Gosling Emacs
- 1985 年 3 月 20 日, Version 13 作为第一个 public release
- Stallman 维护至 2008 年, 现在的维护者是 Stefan Monnier 和 Chong Yidong

介绍

一点历史

- 1970 年, (Editing MACroS) 诞生于 MIT 人工智能实验室
- 1984 年, Stallman 开发 GNU Emacs 用来代替 Gosling Emacs
- 1985 年 3 月 20 日, Version 13 作为第一个 public release
- Stallman 维护至 2008 年, 现在的维护者是 Stefan Monnier 和 Chong Yidong
- 2009 年, 23.1 发布, 支持 Xft

介绍

一点历史

- 1970 年, (Editing MACroS) 诞生于 MIT 人工智能实验室
- 1984 年, Stallman 开发 GNU Emacs 用来代替 Gosling Emacs
- 1985 年 3 月 20 日, Version 13 作为第一个 public release
- Stallman 维护至 2008 年, 现在的维护者是 Stefan Monnier 和 Chong Yidong
- 2009 年, 23.1 发布, 支持 Xft
- 最新版本 23.3

GPL 授权，GNU 项目的第一个计划。

介绍

特性

介绍

特性

- 编辑器：高亮，各种语言支持

介绍

特性

- 编辑器：高亮，各种语言支持
- 高可定制，高可扩展（Emacs Lisp）

介绍

特性

- 编辑器：高亮，各种语言支持
- 高可定制，高可扩展（Emacs Lisp）
- 各种扩展（Mode）：邮件，日历，GTD，IRC

介绍

特性

- 编辑器：高亮，各种语言支持
- 高可定制，高可扩展（Emacs Lisp）
- 各种扩展（Mode）：邮件，日历，GTD，IRC
- 支持本地球上大多数操作系统

介绍

特性

- 编辑器：高亮，各种语言支持
- 高可定制，高可扩展（Emacs Lisp）
- 各种扩展（Mode）：邮件，日历，GTD，IRC
- 支持本星球上大多数操作系统
- Self-documenting

介绍

特性

- 编辑器：高亮，各种语言支持
- 高可定制，高可扩展（Emacs Lisp）
- 各种扩展（Mode）：邮件，日历，GTD，IRC
- 支持本星球上大多数操作系统
- Self-documenting
- 还可以煮咖啡：CoffeeMode

Emacs VS vi

...endless...

拜 Emacs 教

To join the Church of Emacs, you need only say the Confession of the Faith three times:

**There is no system but GNU,
and Linux is one of its kernels.**

学习 Emacs

基本概念

```
# -*- coding: utf-8 -*-
#
# Copyright 2010 yinhm
import os
import sys

if 'vendor' not in sys.path:
    # Add lib as primary libraries directory, with fallback to lib/dist
    sys.path[0:0] = ['lib', 'vendor']

from tornado.web import url

import tornado.wsgi
from google.appengine.ext.webapp.util import run_wsgi_app

# Application handlers
from account.handlers import *
from flog.handlers import *
from recipe.handlers import *
from plan.handlers import *
from about.handlers import *
from handle_task import *
```

1 Buffer

Mini buffer

Major mode

Minor modes

-U:--- main.py 1% L16 Git-ndb (Python Rope AC yas Fill)-----
Find file: ~/workspace/cookio/ | templates/ | LICENSE-Apache | LICENSE-BSD | README.txt | Rakefile | about/ | account/ | app.yaml | appengine_config.py | bin/ | ...}

基本概念

Buffer

基本概念

Buffer

- 持有正在编辑的文件

基本概念

Buffer

- 持有正在编辑的文件
- 打开文件创建对应的文件关联 Buffer

基本概念

Buffer

- 持有正在编辑的文件
- 打开文件创建对应的文件关联 Buffer
- 启动 Emacs 自动打开 `*scrach*` Buffer，不和文件关联

基本概念

Buffer

- 持有正在编辑的文件
- 打开文件创建对应的文件关联 Buffer
- 启动 Emacs 自动打开 `*scrach*` Buffer，不和文件关联
- 启动 Emacs 自动打开名为 `*Messages*` Buffer，类似 STDOUT

基本概念

Buffer

- 持有正在编辑的文件
- 打开文件创建对应的文件关联 Buffer
- 启动 Emacs 自动打开 `*scrach*` Buffer，不和文件关联
- 启动 Emacs 自动打开名为 `*Messages*` Buffer，类似 STDOUT
- 察看文档打开特殊点的 `*Help*` Buffer

基本概念

Buffer

- 持有正在编辑的文件
- 打开文件创建对应的文件关联 Buffer
- 启动 Emacs 自动打开 `*scrach*` Buffer，不和文件关联
- 启动 Emacs 自动打开名为 `*Messages*` Buffer，类似 STDOUT
- 察看文档打开特殊点的 `*Help*` Buffer
- 可以列出、选择、关闭 Buffers

基本概念

Minibuffer

基本概念

Minibuffer

- 输入命令区域

基本概念

Minibuffer

- 输入命令区域
- 命令读取参数

基本概念

Minibuffer

- 输入命令区域
- 命令读取参数
- 文件列表选择

基本概念

Minibuffer

- 输入命令区域
- 命令读取参数
- 文件列表选择
- 可以 TAB 补全

基本概念

Minibuffer

- 输入命令区域
- 命令读取参数
- 文件列表选择
- 可以 TAB 补全
- 可以编辑

基本概念

Minibuffer

- 输入命令区域
- 命令读取参数
- 文件列表选择
- 可以 TAB 补全
- 可以编辑
- Minibuffer 历史

基本概念

Major mode

基本概念

Major mode

- 只能有一个

基本概念

Major mode

- 只能有一个
- 确定主要编辑模式

基本概念

Major mode

- 只能有一个
- 确定主要编辑模式
- python, ruby, lisp, yaml 等

基本概念

Minor modes

基本概念

Minor modes

- 配合 Major mode 可以有多个

基本概念

Minor modes

- 配合 Major mode 可以有多个
- 提供辅助编辑功能

基本概念

Minor modes

- 配合 Major mode 可以有多个
- 提供辅助编辑功能
- 比如 auto-complete, yasnippet

用 Emacs，需忘记鼠标，无视菜单

快捷键控制一切

EMACS = ?

EMACS =
Escape-Meta-Alt-Control-Shift

快捷键

标记方式

快捷键

标记方式

- [Ctrl] 键以大写 C 表示, [Meta] 键 [Alt] 键以大写 M 表示

快捷键

标记方式

- [Ctrl] 键以大写 C 表示，[Meta] 键 [Alt] 键以大写 M 表示
- C-f 表示：按住 CTRL 键再输入 f

快捷键

标记方式

- [Ctrl] 键以大写 C 表示, [Meta] 键 [Alt] 键以大写 M 表示
- C-f 表示: 按住 CTRL 键再输入 f
- C-x-b 表示: 按住 CTRL 键再输入 x, 释放, 输入 b

快捷键

标记方式

- [Ctrl] 键以大写 C 表示, [Meta] 键 [Alt] 键以大写 M 表示
- C-f 表示: 按住 CTRL 键再输入 f
- C-x-b 表示: 按住 CTRL 键再输入 x, 释放, 输入 b
- C-x C-s 表示: 按住 CTRL 键再输入 x, 释放, 按住 CTRL 后输入 s

快捷键

文件操作

快捷键

文件操作

Keystroke	Emacs Command Name	Standard Name
C-x C-f	find-file	打开...

快捷键

文件操作

Keystroke	Emacs Command Name	Standard Name
C-x C-f	find-file	打开...
C-x k	kill-buffer	关闭

快捷键

文件操作

Keystroke	Emacs Command Name	Standard Name
C-x C-f	find-file	打开...
C-x k	kill-buffer	关闭
C-x C-s	save-buffer	保存

快捷键

文件操作

Keystroke	Emacs Command Name	Standard Name
C-x C-f	find-file	打开...
C-x k	kill-buffer	关闭
C-x C-s	save-buffer	保存
C-x C-c	save-buffers-kill-terminal	保存并退出

快捷键

文件操作

Keystroke	Emacs Command Name	Standard Name
C-x C-f	find-file	打开...
C-x k	kill-buffer	关闭
C-x C-s	save-buffer	保存
C-x C-c	save-buffers-kill-terminal	保存并退出
C-x ←	previous-buffer	前一个 Tab

快捷键

文件操作

Keystroke	Emacs Command Name	Standard Name
C-x C-f	find-file	打开...
C-x k	kill-buffer	关闭
C-x C-s	save-buffer	保存
C-x C-c	save-buffers-kill-terminal	保存并退出
C-x ←	previous-buffer	前一个 Tab
C-x →	next-buffer	下一个 Tab

快捷键

光标控制（1）

快捷键

光标控制（1）

Keystroke	Emacs Command Name	Standard Key
C-v	scroll-up	PageDn

快捷键

光标控制（1）

Keystroke	Emacs Command Name	Standard Key
C-v	scroll-up	PageDn
M-v	scroll-down	PageUp

快捷键

光标控制（1）

Keystroke	Emacs Command Name	Standard Key
C-v	scroll-up	PageDn
M-v	scroll-down	PageUp
C-p	previous-line	↑

快捷键

光标控制（1）

Keystroke	Emacs Command Name	Standard Key
C-v	scroll-up	PageDn
M-v	scroll-down	PageUp
C-p	previous-line	↑
C-n	next-line	↓

快捷键

光标控制 (1)

Keystroke	Emacs Command Name	Standard Key
C-v	scroll-up	PageDn
M-v	scroll-down	PageUp
C-p	previous-line	↑
C-n	next-line	↓
C-b	backward-char	←

快捷键

光标控制（1）

Keystroke	Emacs Command Name	Standard Key
C-v	scroll-up	PageDn
M-v	scroll-down	PageUp
C-p	previous-line	↑
C-n	next-line	↓
C-b	backward-char	←
C-f	forward-char	→

快捷键

光标控制（2）

快捷键

光标控制 (2)

Keystroke	Emacs Command Name
M-f	forward-word

快捷键

光标控制（2）

Keystroke	Emacs Command Name
M-f	forward-word
M-b	backward-word

快捷键

光标控制（2）

Keystroke	Emacs Command Name
M-f	forward-word
M-b	backward-word
C-a	move-beginning-of-line

快捷键

光标控制（2）

Keystroke	Emacs Command Name
M-f	forward-word
M-b	backward-word
C-a	move-beginning-of-line
C-e	move-end-of-line

快捷键

光标控制（2）

Keystroke	Emacs Command Name
M-f	forward-word
M-b	backward-word
C-a	move-beginning-of-line
C-e	move-end-of-line
M-a	backward-sentence

快捷键

光标控制（2）

Keystroke	Emacs Command Name
M-f	forward-word
M-b	backward-word
C-a	move-beginning-of-line
C-e	move-end-of-line
M-a	backward-sentence
M-e	forward-sentence

快捷键

编辑操作

快捷键

编辑操作

Keystroke	Emacs Command Name	Standard Name
C-k	kill-line	

快捷键

编辑操作

Keystroke	Emacs Command Name	Standard Name
C-k	kill-line	
C-M-@	mark-sexp	

快捷键

编辑操作

Keystroke	Emacs Command Name	Standard Name
C-k	kill-line	
C-M-@	mark-sexp	
C-w →	kill-region	剪切

快捷键

编辑操作

Keystroke	Emacs Command Name	Standard Name
C-k	kill-line	
C-M-@	mark-sexp	
C-w →	kill-region	剪切
M-w ←	kill-ring-save	拷贝

快捷键

编辑操作

Keystroke	Emacs Command Name	Standard Name
C-k	kill-line	
C-M-@	mark-sexp	
C-w →	kill-region	剪切
M-w ←	kill-ring-save	拷贝
C-y ←	yank	粘贴

快捷键

编辑操作

Keystroke	Emacs Command Name	Standard Name
C-k	kill-line	
C-M-@	mark-sexp	
C-w →	kill-region	剪切
M-w ←	kill-ring-save	拷贝
C-y ←	yank	粘贴
C-/ ←	undo	撤销

快捷键

其它

Keystroke	Emacs Command Name	Standard Name
M-x	execute-extended-command	执行命令
C-g	keyboard-quit	取消输入

Emacs 速成指南: C-h t

忘了快捷键？

记住这个： **C-h b**

GNU Emacs Reference Card

从 emacs-starter-kit 开始

为什么？

每个人都有一个自己的 Emacs

很好很强大， 很糟很分裂

emacs-starter-kit 是好的开始

emacs-starter-kit

提供什么

emacs-starter-kit

提供什么

- 主要针对动态语言
- 合理的配置
- ELPA 包管理
- 支持多种 Emacs 版本

默认支持语言

emacs-starter-kit

默认支持语言

- lisp
- python
- javascript
- perl
- ...

代码在 **github** 上，很容易用来创建自己的分支，扩展其它各种支持

github 上 Popular Forked 第五名

还是很分裂，尽量少分裂

emacs-starter-kit

我的分支

emacs-starter-kit

我的分支

- 超靓主题（来自 emacs-on-rails mode 作者 dima）
- 移植 debian-start，增强启动控制： `/emacs.d/init.d/`
- ruby-test-mode, rsense 等增强 ruby 支持
- rope, ropemacs 增强 python 支持
- 增强 javascript 支持： flymake
- arc mode
- 更多个人口味...

emacs-starter-kit

补充：从软件仓库安装的包和 Modes

emacs-starter-kit

补充：从软件仓库安装的包和 Modes

- emacs23, emacs23-bin-common, emacs23-common
- emacsen-common
- emacs-goodies-el
- css-mode
- erlang-mode
- ess
- html-helper-mode
- mmm-mode
- org-mode
- python-mode

emacs-starter-kit

常用 mode: ido

emacs-starter-kit

常用 mode: ido

- 自 emacs2.2 开始并入官方发布
- C-x b 进入当前所有 Buffers 列表
- 替换 C-x C-f 打开文件
- Tab 补全, Fuzz
- <http://www.emacswiki.org/emacs/InteractivelyDoThings>

emacs-starter-kit

常用 mode: auto-complete

emacs-starter-kit

常用 mode: auto-complete

- 可视界面
- 轻量高效
- 高可扩展，结合各种补全: rope, rsense, gccsence 等
- <http://cx4a.org/software/auto-complete/>

emacs-starter-kit

常用 mode: yasnippet

emacs-starter-kit

常用 mode: yasnippet

- 代码模版
- 默认支持语言: C, C++, Python, Ruby 以及更多
- 语法受 TextMate 影响

emacs-starter-kit

常用 mode: org

emacs-starter-kit

常用 mode: org

- 可以用来 GTD
- 保存 Notes
- 项目管理

demo

Emacs Lisp(Lisp dialect)

Emacs lisp

常见配置可能用到的

Emacs lisp

常见配置可能用到的

- `nil` and `t`

Emacs lisp

常见配置可能用到的

- `nil` and `t`
- `setq`

Emacs lisp

常见配置可能用到的

- `nil` and `t`
- `setq`
- `quote`

Emacs lisp

常见配置可能用到的

- `nil` and `t`
- `setq`
- `quote`
- `require`

Emacs lisp

常见配置可能用到的

- `nil` and `t`
- `setq`
- `quote`
- `require`
- `hook`

Emacs lisp

常见配置可能用到的

- `nil` and `t`
- `setq`
- `quote`
- `require`
- `hook`
- `local-set-key`

Questions?

Links

- <http://www.gnu.org/software/emacs/>
- <http://en.wikipedia.org/wiki/Emacs>
- http://en.wikipedia.org/wiki/Editor_war
- *St IGNUcius*
- *GNU Emacs Reference Card*
- *emacs starter kit*
- 我的 *emacs starter kit*

About

Created in \LaTeX using the *beamer* class, *TeX Live* and *Emacs*.



Published under the Creative Commons Attribution 3.0 License
<http://creativecommons.org/licenses/by/3.0/>

by @yinhm
<http://yinhm.appspot.com>

Document version March 20, 2011