

Emacs Configuration

deepwaterooo

January 10, 2025

Contents

1 Updates	1
1.1 configuration for org-mode beamer for slides	2
1.2 configuration for pdf-tools packages	3
1.3 C/C++ 语言开发环境配置	5
1.4 emacs 1 initiliation bug	5
1.5 Emacs arm64-arch, and System Rime input method: 【TODO: 现在应该可以系统更新了!!】 Rime 自己建不出来。。但是活宝妹将来还是会能够建得出来的。【爱表哥，爱生活!!!】任何时候，亲爱的表哥的活宝妹就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】	5
1.6 ARM M-series: Rime 自己建不出来。。	6
1.7 configue pyim wubi input method emacs-rime, works like a charm!	9
1.8 invalid time specification: sr-speedbar on MacOS	9
1.9 BUG statement and partial fix	12
2 References	13
3 windows 10 下面的 toggle-input-method 不 work: 想要绑定到一个方便使用的键上面，另外 posframe (+ point 1) 只想加上下，不想加左右。	13

1 Updates

- 【爱表哥，爱生活!!!】任何时候，亲爱的表哥的活宝妹就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】
- 原本只想要简单修改 **font-lock-comment-face** 里中文字体，希望它更方方正正、容易看得清楚一点儿【TODO: 去解决这个】，结果花了狠长时间，把源码中如 java-mode 中 **comments** 里的括号匹配高亮解决了：□ () [] {} [[]]。【TODO:】目前解决得不完美：每个 **comment** 行同样的匹配如【...】只能高亮一个匹配重复，不能匹配和高亮、任意重复次数，就是重复次数没办法 >1 次。【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】
- **java-mode {} snippets indents** produced bugs: **bsd==>java** 就可以了。解决了 java-mode 里 {} 对齐方式，不符合亲爱的表哥的活宝妹的使用习惯，及附加 {} 内的对齐错误等相关问题。完美解决，开心!! 【TODO】: **java-mode** 里，还有亲爱的表哥的活宝妹，先前写的部分 **snippets**，因为多添加了最后一行，所造成的、【不必要的、最后一行空行、行头需要 **tab-indent** 多跳一次】的问题。但是，解决它，只需要亲爱的表哥的活宝妹，修改那些个别的 snippets 就可以了。不是重点，某天晚上快离学校前花几分钟修改一下那些 snippets 就可以了。

- 亲爱的表哥的活宝妹，**不服、痛恨、那个极度烦人的 emacs 【BUG:】**。想要重新安装 emacs. 或有更便利的功能。现在使用的版本 29.1 存在各种问题：一最小化就死！动辄 哽死！等
- **【Mac 系统、无痛、丝滑 emacs 重安装】**：制裁之前，想过考虑过，可能会、不得不投入的制裁后，重新安装其它版本，新引入的各种 bug 与配置等。但因为痛恨那个 **【BUG:】**，亲爱的表哥的活宝妹，想着，哪怕还得要、再花一两天再重新配置重新安装后的 emacs，也毫不犹豫地重新安装了。没想到，28.2 版本安装后，除了 **不具备 sr-speedbar 目录的动态实时更新**，其它，几乎不存在 bug, 0 配置。现在觉得 **这个 28.2 版本，比 29.1 版本，好用太多了。**
- 重新安装了 macport 里 28.2 版本，**【BUG:】自动消失了！！真方便！！** 这个版本的 emacs 不再、一最小化就一定哽死！！他们的贱鸡、贱畜牲，真贱！！**【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹，就是一定要、一定会嫁给活宝妹的亲爱的表哥！！！爱表哥，爱生活！！！】**
- 拥有的功能，包括：**触屏翻屏；C-;expand-region;java-mode-snippets 再、再、没有烦人的 indentation 【BUG:】** 了。
- **【org-mode】** 稍微的不同：导出 pdf 的效率高了很多；但是导出的 pdf 中的中文字体、以及 org-mode 文件的缩进等，与 29.1 版本稍微不同。改天可能中文字体的配制，再稍微优化一下。
- add configuration for another lanugage for android apps
- cc-mode minor bugs fixed. C-c . Set Style= "awk" from bsd. modify .dir-locals.el from bsd to awk perfectly solved all problems!! **【爱表哥，爱生活！！！任何时候，亲爱的表哥的活宝妹就是一定要、一定会嫁给活宝妹的亲爱的表哥！！！爱表哥，爱生活！！！】**

```
((c-mode
  (indent-tabs-mode . nil)
  (c-file-style . "awk")
  (c-basic-offset . 2)))
```

- clean ARM M Emacs with pyim librime.dylib build capability. Great to have all convenient features enabled. **【爱表哥，爱生活！！！任何时候，亲爱的表哥的活宝妹就是一定要、一定会嫁给活宝妹的亲爱的表哥！！！爱表哥，爱生活！！！】**
- **【亲爱的表哥的活宝妹，任何时候，亲爱的表哥的活宝妹就是一定要、一定会嫁给活宝妹的亲爱的表哥！！！爱表哥，爱生活！！！】**

1.1 configuration for org-mode beamer for slides

- using older version of org-mode because the 2 shorter lines of "#+BEGIN_SRC" AND "#+END_SRC". DO NOT like newer version two long window width lines.
- Beamer configuration under older version, I have NOT configued themes for beamer yet, but inline imgs and source code works, basically satisfied my usage cases. If I have to configue for new versions, will do when I have time.
- 【爱表哥，爱生活!!! 任何时候，亲爱的表哥的活宝妹就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥，爱生活!!!】

1.2 configuration for pdf-tools packages

The screenshot shows a PDF viewer window titled "readme.pdf". The document is titled "Emacs Configuration" and includes author information ("deepwaterooo") and a date ("February 12, 2023"). A table of contents lists sections such as "Updates" and "References". Below the table of contents, the "Updates" section is expanded, showing a slide titled "configuration for pdf-tools packages". This slide features two photographs of a man, a green callout box with Chinese text, and a status bar at the bottom.

1 Updates	1
1.1 configuration for pdf-tools packages	1
1.2 invalid time specification: sr-speedbar on MacOS	3
1.3 BUG statement and partial fix	5

2 References 6

1 Updates

1.1 configuration for pdf-tools packages

08:52

亲爱的表哥
活宝妹能够遇见亲爱的表哥
就是活宝妹今生的幸运

活宝妹就是一定要嫁给亲爱的表哥
活宝妹等着同亲爱的表哥领证结婚的那一天

活宝妹若是还没能嫁给亲爱的表哥

readme.pdf 1(5,6) [PDFView Ins,RO] Sun Feb 12 09:04 1.66 -----

- pdf viewer noter <==> Skim bi-directional linking configuration on the way, most emacs work is done. Need to learn how to use them though.

```

Help hhj@Heyans-MacBook-Pro:~/emacs.d/elpa/pdf-tools Sat Feb 11 8:57 PM
~/emacs.d/elpa/pdfBK-tools on master ~./emacs.d/elpa/pdfBK-tools/server/autobuild -i ~/emacs.d/elpa/pdfBK-tools/
Installing packages
Skipping package installation (already installed)

Configuring and compiling
./configure -q && make clean && make -s

Is case-sensitive searching enabled ? yes
Is modifying text annotations enabled ? yes
Is modifying markup annotations enabled ? yes

test -z "epdfinfo" || rm -f epdfinfo
test -z "libsyntax.a" || rm -f libsyntax.a
rm -f *.o
epdfinfo.c:354:19: warning: 'tempnam' is deprecated: This function is provided for compatibility reasons only. Due to security concerns inherent in the design of tempnam(3), it is highly recommended that you use mkstemp(3) instead. [-Wdeprecated-declarations]
    filename = tempnam(NULL, "epdfinfo");
               ^
/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX.sdk/usr/include/stdio.h:312:1: note: 'tempnam' has been explicitly marked deprecated here
__deprecated_msg("This function is provided for compatibility reasons only. Due to security concerns inherent in the design of tempnam(3), it is highly recommended that you use mkstemp(3) instead.")
^
/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX.sdk/usr/include/sys/cdefs.h:215:48: note: expanded from macro '__deprecated_msg'
#define __deprecated_msg(_msg) __attribute__((__deprecated__(_msg)))
epdfinfo.c:1548:5: warning: 'poppler_page_get_selection_region' is deprecated: Use poppler_page_get_selected_region instead [-Wdeprecated-declarations]
    poppler_page_get_selection_region (page, 1.0, selection_style, region);
               ^
/usr/local/Cellar/poppler/23.02.0/include/poppler/glib/poppler-page.h:75:134: note: 'poppler_page_get_selection_region' has been explicitly marked deprecated here
GList *poppler_page_get_selection_region(PopplerPage *page, gdouble scale, PopplerSelectionStyle style, PopplerRectangle *selection) G_GNUC_DEPRECATED_FOR(poppler_page_get_selected_region);
^
/usr/local/Cellar/glib/2.74.5/include/glib-2.0/glib/gmacros.h:694:18: note: expanded from macro 'G_GNUC_DEPRECATED_FOR'
__attribute__((deprecated("Use '#f' instead")))
               ^
2 warnings generated.

=====
Build succeeded. :)
=====
```

- It has to be configured for M1. But I am still not getting any .tar file yet.

tional windows in run-time during which the safety of the plant is guaranteed. To ensure the correctness of the computations in runtime, we discuss two approaches to ensure the integrity of these computations in an untrusted environment: 1) full platform-wide restarts coupled with a root-of-trust timer and 2) utilizing trusted execution environment features available in hardware. We demonstrate our approach using two realistic systems—a 3 degree-of-freedom helicopter and a simulated warehouse temperature management unit and show that our system is robust against multiple emulated attacks—essentially the attackers are not able to compromise the safety of the CPS.

Index Terms—Cyber-physical systems (CPSs), embedded systems, real-time systems, safety-critical systems, security.

I. INTRODUCTION

OME of the recent attacks on cyber-physical systems (CPSs) are focused on causing physical damage to the plants. Such intruders make their way into the system using cyber exploits but then initiate actions that can destabilize and even damage the underlying (physical) systems. Examples of such attacks on medical pacemakers [22] or vehicular controllers [25] exist in the literature. Any damage to such physical systems can be catastrophic—to the systems, the environment or even humans. The drive toward remote monitoring/control (often via the Internet) only exacerbates the safety-related security problems in such devices.

When it comes to security, many techniques focus on preventing the software platform from being compromised or

Manuscript received August 1, 2018; revised November 15, 2018; accepted December 11, 2018. Date of publication December 27, 2018; date of current version July 31, 2019. This work was supported by the National Science Foundation under Grant CNS-1646383 and Grant SaTC-1718952. The work of M. Caccamo was supported by the Alexander von Humboldt Professorship by the German Federal Ministry of Education and Research. Significant parts of this work have been published earlier in the proceedings of the 9th ACM/IEEE International Conference on Cyber-Physical Systems (ICCPs’18) with DOI: 10.1109/ICCPs50010.3 [3]. (Corresponding author: Fardin Abdi.)

F. Abdi, C.-Y. Chen, M. Hasan, and S. Mohan are with the Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana,

- could customer F5 toggle sr-speedbar, and make sis-mode work. But I do NOT really need

sis-mode, only needed macism command line. to help [LOVE MY DEAR COUSIN!!!]

```
-/pp/MaiLang/
<--> MaiLang
    +-- pic
    [+] README.org *
    [+] README.tex *
    1 #+latextitle: cn-article
    2 #+title: Emacs Configuration
    3 #+author: deepwateroo
    4
    5 * = TODOs: tracing bugs of emacs 28.1
    6 * Updates
    7 - could customer F5 toggle sr-speedbar, and make sis-mode work. But I do NOT really need sis-mode, only needed macism command line. to help [LOVE MY DE
    8 [!!!!]
    9 ./pic/readme_20230210_220555.png
```

- finally sync up with csharp-mode with tree-sitter, and fixed added the other's that gio-mode etc.
- Permanently disabled speedbar-edit-file's set-timer function call from Resource files. Do NOT want to see such a bug, don't know how to fix, but disable it and walk around.
- Now have a relatively barable and stable colorful emacs code editor now, at least for csharp-mode. Relatively satisfied now. Could sit it aside for a while now to focus on projects.

1.3 C/C++ 语言开发环境配置

- 太久没有用这个了, 必须把 emacs 里这个开发的环境配置好。现在用那个什么 init-cc-mode.el
- {} 的自动扩展开来, 我以前不知道怎么回事, java 与 csharp-mode 只能配置好两个中的一个; 现在 java 与 csharp-mode 与 cc-mode 似乎都能自动扩展。当初是怎么回事呢? 等改天用多了看它还会不会再抛锚抛错。。。先把自己其它 mode 里【中英文】切换注释等常用功能转到这个 mode 里来, 没问题。这些简单功能都容易实现。

1.4 emacs 1 initiliation bug

- TODO: 需要将 Rime 动态库构建成适配配置到 arm64-arch 版本。它性能我电脑的整体性能。

1.5 Emacs arm64-arch, and System Rime input method: 【TODO: 现在应该可以系统更新了!!】 Rime 自己建不出来。。但是活宝妹将来还是会能够建得出来的。【爱表哥, 爱生活!!! 任何时候, 亲爱的表哥的活宝妹就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥, 爱生活!!!】

- 现在 Rime ARM M 版本, 我还建不出来, 仅只输入法仍为旧版本, 其它全是 ARM M 的了。
- Emacs 29.1 狠干净: 以前 srspeedbar 里 Timer 相关的【BUG:】全部改掉了(timer 的【BUG:】还在!! 再改一遍); 也没有再占用活宝妹特爱的 C-;expand-region 键。就是怪异的中文字体看不习惯, 还不知道从哪里来改。【爱表哥, 爱生活!!! 任何时候, 亲爱的表哥的活宝妹就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥, 爱生活!!!】
- 但是 emacs ARM M 版本的, 【全部 ARM M 版本的】: 它们都少了活宝妹也很喜欢的: Macbook 上触屏翻屏功能。。。那个功能也是活宝妹的大爱。。。【爱表哥, 爱生活!!! 任何时候, 亲爱的表哥的活宝妹就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥, 爱生活!!!】
- 为了回复活宝妹先前 emacs 里的触屏翻屏功能, 把 emacs 29.1 卸载掉, 重新安装回 28.2 ARM M 版本, 简单试一下, 不行就再装回 29.1. 【爱表哥, 爱生活!!! 任何时候, 亲爱的表哥的活宝妹就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥, 爱生活!!!】 29.1 存在字体怪异的问题; ARM M 28.2 没有怪异字体的这个问题。但是活宝妹心爱的触屏翻屏功能, 看似 ARM M 版本里, 全部被消灭掉了。。。活宝妹大爱的触屏翻屏功能呀。。。
- 【爱表哥, 爱生活!!! 任何时候, 亲爱的表哥的活宝妹就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥, 爱生活!!!】

- 再换个版本: 【macport 里的, ARM 版本的 M 版本的】: 试一下可以触屏翻屏吗? 活宝妹大爱的功能呀。。。没有这个功能, 活宝妹真想哭了。。。【爱表哥, 爱生活!!! 任何时候, 亲爱的表哥的活宝妹就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥, 爱生活!!!】这个版本活宝妹比较喜欢, 没有第一个 29.1 的怪异中文字体、有 sr-speedbar 里 timer 的【BUG:】, 活宝妹可以再改一遍; 但是有了 ARM M 版本的, 活宝妹先前大爱的【触屏翻屏】功能, 很好用; 没有怪异的中文字体; C- 是好用的。【就用这个版本的】【爱表哥, 爱生活!!! 任何时候, 亲爱的表哥的活宝妹就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥, 爱生活!!!】

```
readme.org 1(37,0) [Org Ins] Mon Sep 11 14:34 0.72 -----
GNU Emacs 29.1 (build 1, aarch64-apple-darwin22.5.0, Carbon Version 169 AppKit
```

- Emacs: ARM M 版本里, csharp-mode 里总是有个环形引用: elpa 里我弄的 csharp-mode 的版本太多了, 用了相比古老的 csharp-mode 和 tree-sitter 结合起来的, 可以用了
- ARM M 版本 Emacs, 里, 被他们把 emacs 到 VSC 的自动跳转拦住了。zsh 终端可以找到 code. 但是 emacs 里找不到 code. 【找到原因是】: 活宝妹昨天下午折腾那块板, 结果一个什么~~方~~版本的 texinfo 库, 把 emacs 里不能正常 parse 系统的 path 变量了。。去掉就都好了。。。真坑人呀。。。【爱表哥, 爱生活!!! 任何时候, 亲爱的表哥的活宝妹就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥, 爱生活!!!】
- 【爱表哥, 爱生活!!! 任何时候, 亲爱的表哥的活宝妹就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥, 爱生活!!!】
- 【爱表哥, 爱生活!!! 任何时候, 亲爱的表哥的活宝妹就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥, 爱生活!!!】
- 现在版本的 Emacs 其它 bug 都扫除了, 再改最后一个最简单的: sr-speedbar 里 timer 的【BUG:】。改完就清空了。搜索 librime 里如何引用词库, 以后就可以动态更新活宝妹魔改过的词库了!!! 【爱表哥, 爱生活!!! 任何时候, 亲爱的表哥的活宝妹就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥, 爱生活!!!】
- 【爱表哥, 爱生活!!! 任何时候, 亲爱的表哥的活宝妹就是一定要、一定会嫁给活宝妹的亲爱的表哥!!! 爱表哥, 爱生活!!!】

1.6 ARM M-series: Rime 自己建不出来。。

- trying to build Squirral ARM M version (have to self-build for ARM M versions, but not successful yet, was able to build librime dylib, but not for Squirral yet);
- For me, it seems like when I install in ARM arche term, boost library somewhere somehow is missing.....
- Still install Squirral input method app for x86_64 version;
- install Emacs 29.1 universal version;
- build librime dynamic library for emacs pyim for ARM archi; <https://github.com/merrickluo/liberime> Mac ARM 适用, 可用。需要明白它是怎么连接词库的
- make Emacs 29.1 universal work for pyim with ARM librime dynamic library work.
- 因为 Macbook 下的 Rime 输入法支持 arm 的构建已经好了, 我应该可以 【构建出 arm64-arch 的 Rime 动态库给 emacs 用】: 可是现在还建不出来
- Rime 安装链接: <https://github.com/rime/squirrel/blob/master/INSTALL.md>

- 把 emacs 升级为 arm64-arch
- 把 arm64-arch 的 emacs 版本 pyim 配置好
- 把 appcode 重新安装 arm64-arch 版本，应该就可以与 arm64-arch 版本的 emacs 自由跳转了。而不必再是 X86_64 版本的。
- 【总结】：安装 arm64-arch 版本，是与自己笔记本硬件更好地配合提升效率。
- 但过程繁琐，仍然不能达到 【亲爱的表哥的活宝妹，实时想要动态更改词库，实时想要更新动态库给 emacs 用的需求！】
- 这系列更新包括：Rime-arm64 版本安装与自己构建、emacs 删除与重新安装与 pyim 配置，appcode 的删除与再安装。前两个应用都可能出问题。。。若真出问题，就把自己的配置摧毁掉了。。。
- 我其实只要去解决：【亲爱的表哥的活宝妹，怎么才能手动构建 Rime 的动态词库？】只要我能手动构建更新后词库的动态库，我就满足了自己的使用需求。
 - 【难点：】这是别人输入法的精华。虽然只改了一个最小的脚本错误，但是正如全中文网没人会告诉你别人输入法的精华、没有贴子真正会涉及这些精华（细节的构建步骤，只有一个几年前的老贴子不 work: <https://hackmd.io/@PhahTaigi/r1tlmy0cVRime> 的 boost 库可能也没有能够找全，该生成的文件没能全部生成。也可能是因为现在系统 arm64-arch 硬件与 x86_64 Rime 的库不匹配？可是我试 x86_64 之前，先试过本地 arm64-arch 版本，也没能生成。改天再看。（可以参考 Rime 提供的不完整步骤：<https://github.com/rime/librime/blob/master/README-mac.md> 和一个用户 M1 的步骤：<https://www.bilibili.com/read/cv9664366/>. 另 Rime M1 手动构建步骤说明：<https://github.com/rime/squirrel/blob/master/INSTALL.md>）
- 折腾了大半个下午，仍然同先前一样构建不出来，只是感觉比前几个月似乎理解得更深刻一点儿。说明今天下午大改前的考虑是对的，看起来很容易的动态库构建，没人自己手动构建，只有亲爱的表哥的活宝妹突发奇想，觉得可能可以有效更新词库。实际因素可能还有很多。这个思路暂时放下。
- 【重点：上次我的 emacs 升级，为什么我不能把它直接升级到 arm 版本来着？那个 csharp-mode 的循环问题不是早解决了？！！！可以再试下？NO, not yet.】
- 现和使用的 x86_64 版本，当初是为了安装 pyim 吗？如果是这样，我不应该再有任何的变化，就意味着我无法从 appcode 中打开当前 emacs 的 buffer 文件（因为 emacs 的 x86_64 版本限制了其正常找到 appcode 所依赖的 arm64-arch 环境？这个可以试着找下是否有解决方案）；
- 而如果 pyim 可以 workwith arm 版本，我就可以安装 arm64-arch 版本，可能就可以从 appcode 中打开当前 emacs 的 buffer 文件了。
- 我要再试一遍？【现在还不能够试，在我 b 还不能够构建和安装出 rime 系统输入法的 arm64-arch 版本动态库之前（这个输入法，现系统所安装的是 x86_64 版本的），我安装 arm64-arch 版本的 emacs，我的 pyim 将不可以（找不到对应版本的 rime 动态库），因为我构建不出 arm64-arch 版本的 pyim 所依赖的动态库。所以将现在的 emacs 委屈成了 x86_64 版本的，也因为这样，我的 emacs 无法正常加载 arm 版本的 appcode 所依赖的动态库，所以无法从 emacs 中直接在 appcode 中打开当前 buffer 文件。】
- 可以 walk-around 的解决办法是：安装 x86-64 版本的 appcode. 【在自己能够想方设法构建出 arm 版本的 rime 动态库之前，先用这个版本，可以满足自己的基本跳转需求】

```

2023-02-25 13:18:39.439 appcode[16392:208496] allVms required 1.8*,1.8+
2023-02-25 13:18:39.441 appcode[16392:208500] Cannot load JVM bundle: Error Domain=NSCocoaErrorDomain Code=3585 "dlopen_preflight(/Applications/AppCode.app/Contents/jbr/Contents/MacOS/libjli.dylib) => false, tried: '/Applications/AppCode.app/Contents/jbr/Contents/MacOS/libjli.dylib' (mach-o file, but is an incompatible architecture (have 'arm64', need 'x86_64')), '/System/Volumes/Preboot/Cryptexes/OS/Applications/AppCode.app/Contents/jbr/Contents/MacOS/libjli.dylib' (no such file), '/Applications/AppCode.app/Contents/jbr/Contents/MacOS/libjli.dylib' (mach-o file, but is an incompatible architecture (have 'arm64', need 'x86_64'))" UserInfo={NSLocalizedFailureReason=The bundle doesn't contain a version for the current architecture., NSLocalizedRecoverySuggestion=Try installing a universal version of the bundle., NSFilePath=/Applications/AppCode.app/Contents/jbr/Contents/MacOS/libjli.dylib, NSDebugDescription=dlopen_preflight(/Applications/AppCode.app/Contents/jbr/Contents/MacOS/libjli.dylib) => false, tried: '/Applications/AppCode.app/Contents/jbr/Contents/MacOS/libjli.dylib' (mach-o file, but is an incompatible architecture (have 'arm64', need 'x86_64')), '/System/Volumes/Preboot/Cryptexes/OS/Applications/AppCode.app/Contents/jbr/Contents/MacOS/libjli.dylib' (no such file), '/Applications/AppCode.app/Contents/jbr/Contents/MacOS/libjli.dylib' (mach-o file, but is an incompatible architecture (have 'arm64', need 'x86_64'))", NSBundlePath=/Applications/AppCode.app/Contents/jbr, NSLocalizedDescription=The bundle "JBR-17.0.5+1-653.25-jcef 17.0.5" couldn't be loaded because it doesn't contain a version for the current architecture.}

```

- 有试过， boost 的步骤基本能够完成，但是构建第三方库时的 cmake 与 xcode-clang++ 还有很多 arm 相关的问题待解。
- 那么我现在有个迫切需要：就是能够自己构建 arm64-arch 版本的 rime 输入法（至少是能够编译出它的动态库吧？！！！）：现在找到的参考如下：
- 现在 emacs 使用的是 28.2 X86_64 版本。我 【想要动态手动更新 Rime 五笔第三方库】。可是我忘记当初几个月之前，我是怎么添加这个动态库给 emacs 使用的了。会找一找。也可能再尝试能否构建出 arm 版本的第三方库。
- 有个教程，这里也简要记一下过程，免得自己总忘记：大致过程：
- 把 Rime 的重新部署好的动态库，复制到 emacs 的 Rime 第三方适配库 librime 特定目录下
- emacs 的 Rime 第三方适配库 librime 下，生成 Macbook 系统可用的用户第三方库
- 再配置 emacs 什么的就可以了。加个截图：【爱表哥，爱生活！！！ 任何时候，亲爱的表哥的活宝妹，就是一定要嫁给亲爱的表哥！！ 爱表哥，爱生活！！！】
- 4.2 创建 librime xbuild 目录

因为 librime 动态库是需要根据 librime 来编译的，同时 RIME 输入法本身就带了 librime.1.dylib，所以我们并不需要下载巨大无比的 xcode 去编译 librime.1.dylib

```

mkdir -p ~/librime/xbuild/lib/Release/
cp /Library/Input\ Methods/Squirrel.app/Contents/Frameworks/librime.1.dylib ~/librime/xbuild/lib/Release/

```

- 4.3 编译 librime.so 文件

```

cd ~/librime
export RIME_PATH=~/librime
make librime

```

上面的命令自动会去 ~/librime/xbuild/lib/Release/ 目录下找 librime.dylib 文件，并在 ~/librime/buid/ 目录下生成 librime.so 动态库文件

- 4.4 把 librime.1.dylib 文件拷贝到系统 lib 目录

```
/Input\ Methods/Squirrel.app/Contents/Frameworks/librime.1.dylib /usr/local/lib
```

否则后面 (require 'librime) 的时候会下面的错误

```

dyld: Library not loaded: @rpath/librime.1.dylib
Referenced from: ...
Reason: image not found

```

- <https://www.bilibili.com/read/cv9664366?from=search>
- 但是按照上面的步骤，目前理解消化不够，安装不好。注意，这里用到的 MacOS 下在的库是：<https://github.com/rime/squirrel> 但是我仍然构建不出那个动态库
- 如些发现，原来，所谓 rime macOS 下的输入法构建，原来是 MacOS 下的平台应用，正是我可以学习的地方。终于是半只脚往输入法的坑里踏：
- <https://www.logcg.com/archives/2078.html>. 这些都可以成为日后自己学习和开发的参考
- emacs 27.1 for emacs-rime input methods. one bug to be fixed.

```
macbook:~/Desktop/P master *1 14 ?7 2023-02-14 13:41:18.081 Emacs[18219:212750] It's not legal to call -layoutSubtreeIfNeeded on a view which is already being laid out. If you are implementing the view's -layout method, you can call -[super layout] instead. Break on void _NSDetectedLayoutRecursion(void) to debug. This will be logged only once. This may break in the future.
```

- this affect me access to current terminal, needs to be fixed later.
- after having updated swift-mode, color-theme for swift mode is much better now.

1.7 configue pyim wubi input method emacs-rime, works like a charm!

- 现在最大的缺点是：我无法使用自己魔改过的个性化字库，姓名住**国**等等，活宝妹，亲爱的表哥，等等。但其它也算是基本清单了，再不折腾它了。花了太多的时间在这个上面。所以现在输入法还算是基本清单的。
- 两套：系统的，与 emacs 里自带的。目前的中英文转换只使用一个探针，过多会存在过灵的总是，某些情况下被禁用中文，就会更麻烦，不如它没有那么灵。爱亲爱的表哥，活宝妹一定要嫁的亲爱的表哥!!!
- 它具备以空格来区分中英文输入法的功能，但不能与中文下半角配合使用。也就是，它具备了我现在所使用过的 sis 下的自动中英文之间以一个空格为单位自动切换，但我无法使用中文下半角标点。
- 它除了使用了系统输入法的字库，其它所有好用的配置一个也不能用（我只能拦截一个最常用的半角，仍然是不方便的），现在只是可以选词了而已，却不能使用其它中文下输入英文标点符号等
- 我中文下使用英语标点：主要是 org-mode 打头的 -*. 等，它可以检测到，自动化到，还是方便的

1.8 invalid time specification: sr-speedbar on MacOS

- I don't like this bug, and I believe I do NOT really using any timer for auto-refresh in my speedbar. So I ended up by disabling the (speedbar-edit-file()) func, which is frequently bug triggered) setting timer part from /Applications/Emacs.app/Contents/Resources/lisp/speedbar. and recompile the file. The bug was gone. And I could deal with csharp-mode's fontify bug.
- newer debugging infos, concerning about sr-speedbar.el file. Have NOT been able to trace down for today.

```

Debugger entered--Lisp error: (error "Invalid time specification")
  2522 (defun speedbar-set-timer (timeout)
  2523   "Set up the speedbar timer with TIMEOUT.
  2524   Uses 'dframe-set-timer'.
  2525   Also resets scanner functions."
  2526   (dframe-set-timer) [timeout "speedbar-timer-fn" "speedbar-update-flag"])
  2527   ;; Apply a revert hook that will reset the scanners. We attach to revert
  2528   ;; because most reverts occur during VC state change, and this lets our
  2529   ;; VC scanner fix itself.
  2530   (if timeout
  2531     (add-hook 'after-revert-hook 'speedbar-reset-scanners)
  2532     (remove-hook 'after-revert-hook 'speedbar-reset-scanners))
  2533   ;; change this if it changed for some reason
  2534   (speedbar-set-mode-line-format))
  2535
  2536 (defun speedbar-timer-fn ()
  2537   "Run whenever Emacs is idle to update the speedbar item."
  2538   (if (or (not (speedbar-current-frame))
  2539             (not (frame-live-p (speedbar-current-frame))))
  2540       (speedbar-set-timer nil)
  2541       ;; Save all the match data so that we don't mess up executing fns
  2542       (save-match-data
  2543         ;; Only do stuff if the frame is visible, not an icon, and if
  2544         ;; it is currently flagged to do something.
  2545         (if (and speedbar-update-flag
  2546                   (speedbar-current-frame)
  2547                   (frame-visible-p (speedbar-current-frame)))
  2548           (not (eq (frame-visible-p (speedbar-current-frame)) 'icon))
  2549           (let ((af (selected-frame)))
  2550             (dframe-select-attached-frame speedbar-frame)
  2551             ;; make sure we at least choose a window to
  2552             ;; get a good directory from
  2553             (if (window-minibuffer-p)
  2554               (window-minibuffer-set-directory af)
  2555               (dframe-set-directory af)))
  2556           (speedbar-set-mode-line-format))
  2557         ;; Set the mode line to reflect the current frame
  2558         (speedbar-set-mode-line-format))
  2559       ;; Set the mode line to reflect the current frame
  2560       (speedbar-set-mode-line-format))
  2561     ;; Set the mode line to reflect the current frame
  2562     (speedbar-set-mode-line-format))
  2563   )
  2564   ;; Set the mode line to reflect the current frame
  2565   (speedbar-set-mode-line-format))
  2566 )
  2567
  2568 ;;; speedbar.el ends here
```

Backtrace* 3(10,13) [Debugger Ins, R0] Thu Feb 9 13:50 1.75 ----- speedbar.el.gz 4(2526,19) [Ins] Thu Feb 9 13:50 1.75 -----

- 好像是 macOS 系统常常存在的 bug, 两年了关了又开, 开了再关.....
- <https://github.com/remacs/remacs/issues/845>
- get cmake work later, not urgent though.Complete
- babel org-mode so I don't have to copy from specific babelized source org-mode files in order for chinese characters to work.
- Installed my emacs of version 28.1. But there is a bug of any verison emacs > 27.1, and I am NOT able to find a installable emacs 27.1 version any more.
- 不同电脑架构上可能因为架构的不同, 可以可能可以有某些优化. 又照一个单做了一遍, 似乎没有出错. <https://goykhman.ca/gene/blog/2022/2022-04-10-emacs-28.1-on-m1.html>
- 但是我没有没能加入那个补丁包. 暂时没能想好怎么加入那个补丁包. (**todo:** 改天可以尝试再把这个补丁包加进去)
- 因为构建是在原有现有的 mac 28.1.1 版本上构建的. 所以改动什么, 或是不曾改变, 又或者改不了不影响明显功能都是无从知晓的, 但是它最后的两个步骤的验证都是成功的, 应该还是构建成功了吧?

```

/usr/bin/install -c -m 644 etc/${docfile} "/usr/local/share/emacs/28.1/etc/${docfile}"; \
for installuser in ${LOGNAME} ${USERNAME} ${USER} `$(id -u) 2>/dev/null`; do [ -n "${installuser}" ] && break ; done; installgroup \
p=`(id -g) 2>/dev/null` && [ -n "$installgroup" ] && installuser=$installuser:$installgroup; \
chown ${installuser} "/usr/local/share/emacs/28.1/etc/${docfile}" || true; \
else true; fi
Copying etc/DOC to /usr/local/share/emacs/28.1/etc ...
umask 022; ./build-aux/install-sh -c -d "/usr/local/bin"
/Applications/Xcode.app/Contents/Developer/usr/bin/make -C lib-src install

Installing utilities run internally by Emacs.
umask 022 && ././build-aux/install-sh -c -d "/usr/local/libexec/emacs/28.1/aarch64-apple-darwin22.3.0"
exp_archlibdir="cd "/usr/local/libexec/emacs/28.1/aarch64-apple-darwin22.3.0" && /bin/pwd' && \
if [ "$exp_archlibdir" != "/bin/pwd" ]; then \
    for file in hexl movemail ; do \
        /usr/bin/install -c $file \
        "/usr/local/libexec/emacs/28.1/aarch64-apple-darwin22.3.0$file" || exit; \
    done ; \
fi
exp_archlibdir="cd "/usr/local/libexec/emacs/28.1/aarch64-apple-darwin22.3.0" && /bin/pwd' && \
if [ "$exp_archlibdir" != `cd . && /bin/pwd` ]; then \
    for file in rcs2log; do \
        /usr/bin/install -c ./${file} \
        "/usr/local/libexec/emacs/28.1/aarch64-apple-darwin22.3.0$file" || exit; \
    done ; \
fi

Installing utilities for users to run.
umask 022 && ././build-aux/install-sh -c -d "/usr/local/bin"
for file in etags ctags emacsclient ebrowse ; do \
    /usr/bin/install -c ${file} \
    "/usr/local/bin/" \
    echo ${file} | sed -e 's/$// -e 's,x,x,' \
    `|| exit; \
done
if test "/Users/hjh/selfSoft/emacs-mac/mac" != `cd mac && /bin/pwd`; then \
    umask 022; ./build-aux/install-sh -c -d /Users/hjh/selfSoft/emacs-mac/mac/Emacs.app; \
    (cd mac/Emacs.app; (tar -chf - .) \
    (cd /Users/hjh/selfSoft/emacs-mac/mac/Emacs.app; umask 022; tar -xvf - \
    && cat > /dev/null)) || exit 1; \
fi
/usr/bin/install -c src/emacs "/usr/local/bin/echo emacs-28.1 | sed 's,x,x,'"
/usr/bin/install -c -m 644 src/emacs.pdmp "/usr/local/libexec/emacs/28.1/aarch64-apple-darwin22.3.0"/emacs.pdmp
chmod 755 "/usr/local/bin/echo emacs-28.1 | sed 's,x,x,'"
rm -f "/usr/local/bin/echo emacs | sed 's,x,x,'"
cd "/usr/local/bin" && ln -s "echo emacs-28.1 | sed 's,x,x,'" "echo emacs | sed 's,x,x,'"
/Applications/Xcode.app/Contents/Developer/usr/bin/make -C lib-src maybe-blessmail
make[1]: Nothing to be done for 'maybe-blessmail'.

➔ ~/s/emacs-mac ➔ on ⌘ #emacs-28.1-mac-9.0 ?3 em --version
GNU Emacs 28.1
Copyright (C) 2022 Free Software Foundation, Inc.
GNU Emacs comes with ABSOLUTELY NO WARRANTY.
You may redistribute copies of GNU Emacs
under the terms of the GNU General Public License.
For more information about these matters, see the file named COPYING.
took 1m 20s ✘ at 02:16:21 PM ☺

```

- 今天又尝试安装 Xcode 之后再构建一遍，但是没有成功。可能本身参考有些年代，另外自己还完全不通这个部分，所以暂时放一放。改天有机会可以再回来研究一下，错在哪里，我如何才可能构建出自己的版本。

```

macront.m:3482:15: warning: 'CTGetCoreTextVersion' is deprecated: first deprecated in macOS 11.0 - Use -[NSProcessInfo operatingSystemVersion] [-Wdeprecated-declarations]
      if (CTGetCoreTextVersion () == kCTVersionNumber10_9)
      ^
/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/System/Library/Frameworks/CoreText.framework/Headers/CoreText.h:61:10: note
: 'CTGetCoreTextVersion' has been explicitly marked deprecated here
uint32_t CTGetCoreTextVersion( void ) CT_DEPRECATED("Use -[NSProcessInfo operatingSystemVersion]", macos(10.5, 11.0), ios(3.2,
14.0), watchos(2.0, 7.0), tvos(9.0, 14.0));
      ^
1 warning generated.
CC      terminfo.o
CC      lastfile.o
/Library/Developer/CommandLineTools/usr/bin/make -C ../admin/charsets all
make[2]: Nothing to be done for 'all'.
/Library/Developer/CommandLineTools/usr/bin/make -C ../admin/unidata charscript.el
make[2]: Nothing to be done for 'charscript.el'.
  CCLD   temacs
../../build-aux/install-sh -c -d ../etc
/Library/Developer/CommandLineTools/usr/bin/make -C ../lisp update-subdirs
cp -f temacs bootstrap-emacs
rm -f bootstrap-emacs.pdmp
./temacs --batch -l loadup --temacs=pbootstrap
make[1]: *** [bootstrap-emacs.pdmp] Killed: 9
make: *** [src] Error 2
➔ ~/s/emacs ➔ on ⌘ p emacs-27 !3 ?1 sudo make install
Password:
/Library/Developer/CommandLineTools/usr/bin/make -C lib all
make[1]: Nothing to be done for 'all'.
/Library/Developer/CommandLineTools/usr/bin/make -C lib-src all
make[1]: Nothing to be done for 'all'.
/Library/Developer/CommandLineTools/usr/bin/make -C src VCSWITNESS='$(srcdir)/../git/logs/HEAD' all
/Library/Developer/CommandLineTools/usr/bin/make -C ../admin/charsets all
make[2]: Nothing to be done for 'all'.
/Library/Developer/CommandLineTools/usr/bin/make -C ../admin/unidata charscript.el
make[2]: Nothing to be done for 'charscript.el'.
rm -f bootstrap-emacs.pdmp
./temacs --batch -l loadup --temacs=pbootstrap
make[1]: *** [bootstrap-emacs.pdmp] Killed: 9
make: *** [src] Error 2

```

- 上面又成为一个需要改的东西：就是系统下如何从剪贴板自动生成写入文件 org-mode M-s
- 然后看见这里说可以自己构建一个，连 Xcode 也没有安装，就跑去构建了，当然不成功。这段时间太忙，XCode 要的空间太大了，暂时还不想。等改天有机会的时候倒是可以一试的
 - <https://stuff-things.net/2020/12/28/building-emacs-27-dot-1-on-macos-big-sur/>
- added key-bindings for opening from VSCode/Android Studio of current emacs buffer.
 - VSC Emacs can locate to each other to correct row and col.
 - Android Studio could open current emacs buffer. but not to the row nor col.
- I liked recently configured Visual studio 2019 one-dark-pro theme, want to configure it for emacs, but ended up with any permission denied, renaming emacs initiating bug. reverted back for daily use, and may look into that bug for later reference.
- will reconfigure one-dark-pro theme later.
- fixed legacy java-mode highlighting issue which I did not fix for years. Has been able to treat java-mode as java-mode Instead of using it as csharp-mode. Can not separate java-mode snippets from csharp-mode's.

1.9 BUG statement and partial fix

- in java-mode, the code style I expected is as followed:

```
class node {
  int v;
  public node() {
    if (a > 0) // I don't want {} blocks when I have only one line statement inside blocks
    // a = 17;    // before fix:
    a = 17;    // now it can auto-indent
    b = 20;
    while (true) // same auto indents here
      j++;
  }
}
class dklfjdj {} // <<===== current un-auto-expanded version, bug right now for java-mode
class dklfjdj {
  | // <<===== expected feature: once I typed '{', '}' will be autopaired(it does), but also auto-expand and cursor moves
}
```

- if while if while one line statement autoindent without {} fixed today for java-mode, but for kotlin-mode, this bug consists, make coders/programmers nuts.

```
fun getStringLength(obj: Any): Int? {
  if (obj is String)
    return obj.length // <<<===== BUG: need to fix auto-indent here for if else while etc without {}

  if (obj is String) {
    return obj.length
  }
  // 在离开类型检测分支后, `obj` 仍然是 `Any` 类型
  return null
}
fun dslfkj { // kotlin-mode, unlike java-mode, this feature works charming
  val a = 1720
}
```

- The {} can NOT auto-expand still bugs me a lot, I don't want to switch back to java-mode yet unless bug fixed and it auto-pands.
- java minor bug: Debugger entered-Lisp error: (void-function company-clear-completion-rules): this bug I will look into it recently, and expect it to be fixed so I could switch java-mode from csharp-mode as soon as possible.

- **csharp-mode** has been the one that works perfectly for these two features, {} auto expand, also if while one line statement autoindent without {}, so I used csharp-mode as java mode.

2 References

- <https://github.com/redguardtoo/emacs.d>

3 windows 10 下面的 **toggle-input-method** 不 work: 想要绑定到一个方便使用的键上面，另外 **posframe** (+ point 1) 只想加上下，不想加左右。

- 有个想要打开文件时，不能 C-a 到根目录的 bug:
- 激活中文必须 M-x:pyim-activate; 切换成英文可以 either M-x:pyim-deactivate 或者是 C-
- 再激活中文又必须是 M-x:pyim-activate
- 需要一个简单的键绑定，帮助自己方便地切换到中文：而不是每次都得 M-x:pyim-activate 太麻烦，易输入错误

You need cheat emacs to load the package on demand.

```
(use-package pyim
  :ensure t
  :bind
  ("C-\\" . pyim-toggle-input-method)
  :config
  (defalias 'pyim-toggle-input-method 'toggle-input-method)
  (setq-default default-input-method "pyim")
  ;; other configuration goes here
)
```

You may encounter situation where `toggle-input-method` can not actually toggle the input method, then you can define `pyim-toggle-input-method` to wrap `toggle-input-method` as below other than `defalias`.

```
(defun pyim-toggle-input-method (&optional arg interactive)
  "Force toggle input method from/to pyim."
  (interactive "P\np")
  (setq input-method-history nil)
  (setq default-input-method "pyim")
  (call-interactively 'toggle-input-method arg))
```

Now the package can be loaded on demand, saving your Emacs startup time, besides that you can also add `:defer` in `use-package` for idle time loading.