



项目申请书



项目 ID: 2012153

Emacs Application Framework

申请人: *Hollow Man*

最后修改时间: 2020.6.13

目录

1. 介绍	1
A. 关于此项目	1
B. 关于我	2
2. 项目细节	3
A. 任务 1	3
B. 任务 2	3
C. 任务 3	4
D. 任务 4	4
E. 任务 5-7	5
F. 任务 8	5
G. 任务 9	5
H. 任务 10	5
I. 任务 11	5
J. 任务 12	5
K. 任务 13	6
L. 任务 14	6
M. 任务 15	6
N. 备选任务 1	6
O. 备选任务 2	6
3. 时间安排	6
A. 第 1 周 (7.1 - 5, 2020)	7
B. 第 2-6 周 (7.6 - 8.9, 2020) 共计 5 周	7
C. 第 7 周 (8.10 - 15, 2020)	7
D. 中期报告 (8.15, 2020)	7
E. 第 8-13 周 (8.17 - 24, 2020) 共计 5 周半	7
F. 第 13-14 周 (9.25 - 9.30, 2020)	7
G. 结项(9.30, 2020)	8
H. 未来 (10.1, 2020 -)	8

1. 介绍

A. 关于此项目

此项目计划为 EAF 框架做出贡献，进一步完善和扩展其功能。

根据项目产出要求，即完成清单中所有低难度项目和 7 个中等难度的项目以及 1 个高难度项目，并结合我的意向和特长，我选择以下任务作为我的申请项目目标。在实际工作过程中，可能会出现部分任务过于复杂或者是无法达到预期目标的情况，因而我还另外选择了两个备选任务作为补充：

序号	名称	难度
1	Core:研究支持 chemacs 的可能性。如果可能，请支持，否则请解释。 # 297	低
2	Core:所有功能的英文文档	低
3	Terminal:使支持用 EXIT 信号关闭 Terminal 缓冲区	低
4	Terminal:使支持在 node-pty 中捕获更改目录信号，重命名终端缓冲区并实时更改 Emacs 默认目录变量	低
5	Mindmap:使支持从颜色列表中选择，更改节点颜色	低
6	Mindmap:使支持多行插入 Dons 节点	低
7	Mindmap:使支持以正确的缩进级别导出 org	低
8	Core: 使用 QTest::QTouchEventSequence 模拟触摸滚动事件并控制所有浏览器基础应用	中
9	Browser:使支持通过将密码存储在数据库中来自动填充密码	中
10	Browser:使支持使用光标智能选择文本区域	中
11	Browser:使支持无鼠标浏览。# 270	中
12	Terminal: 使支持服务器准备就绪后刷新页面，因为有时终端在打开时为空白	中
13	Terminal:修复连接建立错误： net::ERR_CONNECTION_REFUSED	中
14	PDF:为 PDF 页面发明更好的暗模式颜色算法	中
15	PDF:为 PDF 页面发明一种更好的暗模式颜色算法并且排除图像部分	高

备 1	Painter: 使用 Qt 或 JavaScript 创建此新应用	中
备 2	RSS Reader: 用 JavaScript 重写 RSS Reader	中

B. 关于我

我叫Hollow Man，这学期参与了留学基金委优秀本科生国际交流项目，目前正在英国交流。

我热爱开源的精神，这就是我加入并积极参与兰州大学开源社区活动的原因。我认为开源最吸引人的地方在于，来自不同背景的人跨越不同的时间线无国界地工作，以实现一个共同的目标，即便他们的项目没有任何回报。

我一直热衷于编程。我在 Github 上使用各种编程语言建立了许多自己的小规模项目存储库。然而，有一件事我还没有经历过，那就是在一个大型的开源组织中工作。尽管大多数学生会选择 BAT 这样的大厂进行暑期实习，但我认为这只是一个常规的单调实习，因为一旦实习期结束，我们与公司的关系就会结束，我永远不会知道我写的代码是如何被使用的。因此，我决定为一个开源社区做出贡献，与来自全国甚至世界各地的开发人员合作，这就是为什么我选择开源软件供应链点亮计划作为我的暑期实习。

我选择为 EAF 做贡献，不仅仅是因为我认为我对 Python、JavaScript 和 PyQt5 比较熟悉，还因为我对 Emacs 有足够的兴趣和激情。目前通过查看相关开发文档和资料，我已经对 DBus API 和 Emacs 有了一定的了解，因为我对 Conjure 语言有一些了解，所以学习起同样是 Lisp 方言的 Emacs Lisp 对我来说是得心应手的。我曾经使用 PyQt5 和 Python 实现了基于离散小波变换的图片信息隐藏加解密软件（地址：<https://github.com/HollowMan6/Hollow-Software-for-Picture-Video-Information-Hiding-Encryption-and-Decryption>，软著号：2020SR0079950），还使用 JavaScript 制作出了自己的智慧树和超星尔雅课程助手（刷课）插件（地址：<https://github.com/HollowMan6/ChaoXing-ErYa-Wisdom-Tree-Lessons-Helper>）。另外在申请期间，我已经迫不及待地为 EAF 做出了自己的贡献：实现了反转模式下图片部分的排除：<https://github.com/manateelazycat/emacs-application-framework/pull/308>，完成了对 Chemacs 的支持性考察：<https://github.com/manateelazycat/emacs-application-framework/pulls/309>。

——如果我的提案和努力被接受，最终成功地融入上游代码库，我会为自己感到骄傲。

2. 项目细节

下面我将分别简单介绍一下到目前为止我对每个任务的理解和实行方案。任务序号对应关系请参见项目介绍部分的表格。

A. 任务 1

（已经解决）

此任务要求研究支持 Chemacs 的可能性。

Chemacs: <https://github.com/plexus/chemacs> 是一个 Emacs 配置文件切换器，它使同时运行多个 Emacs 配置变得容易。可以将其理解为 Emacs 的“操作系统启动器”，使用时只需要在 emacs 启动命令加上 “--with-profile [你的配置名称]” 参数即可进行操作。如果不加该参数启动的是默认配置文件。

更多 Chemacs 用法请参照 Chemacs 官方说明，此处不再进行叙述。

将 EAF 改造的适用于 Chemacs，只需要在代码库根目录增加一个 init.el 文件，内容如下：（代码摘自 <https://manateelazycat.github.io/eaf/emacs/2020/05/10/eaf-community.html> “EAF 安装” 的第三部分 “加载 EAF”）

```
(defun add-subdirs-to-load-path (dir)
  "Recursive add directories to `load-path'."
  (let ((default-directory (file-name-as-directory dir)))
    (add-to-list 'load-path dir)
    (normal-top-level-add-subdirs-to-load-path)))
(add-subdirs-to-load-path "~/ .emacs.d/")

(require 'eaf)
```

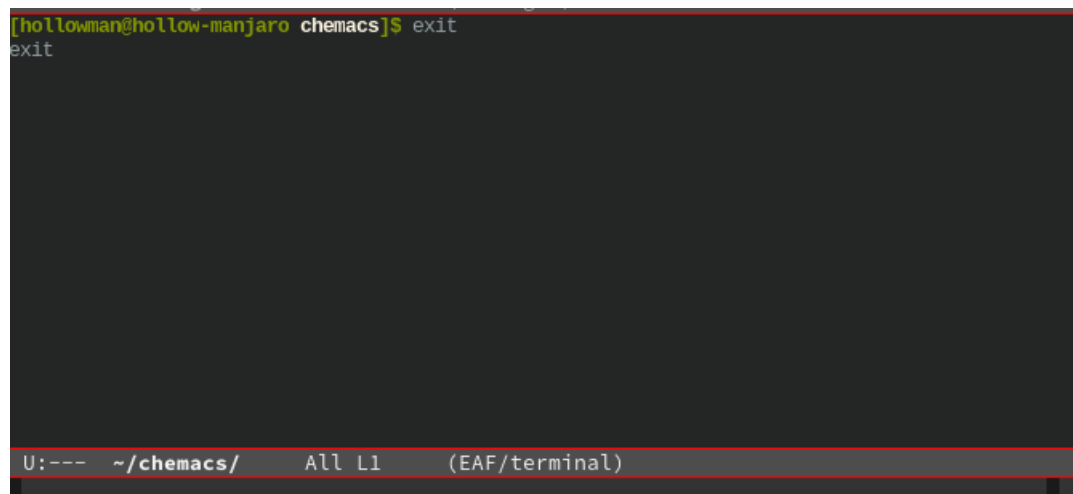
用户如果不想将 EAF 放在 “~/ .emacs.d/” 根目录处，只需要自定义上述红线部分为自己的安装路径，并修改自己的 Chemacs 配置即可。

B. 任务 2

此任务要求编写所有功能的英文文档，所以我将会在已有文档的基础上，通过查看源代码，编写所有功能即 “Browser、HTML Email Renderer、PDF Viewer、Video Player、Image Viewer、Markdown Previewer、Org Previewer、Camera、Terminal、File Sender、File Browser、Airshare、RSS Reader、Mindmap、Doc Viewer、Mermaid” 功能的英文文档。这个任务我会放在第一个做，从而加深我对 EAF 的了解度，为后续的编程做准备。

C. 任务 3

此任务要求用 EXIT 信号关闭 Terminal 缓冲区。Emacs 关闭缓冲区的办法是 C-x-k (Ctrl+x k)，目前在 terminal 中输入 exit 只会使得此次对话结束，并没有智能地关闭此 Terminal 缓冲区：



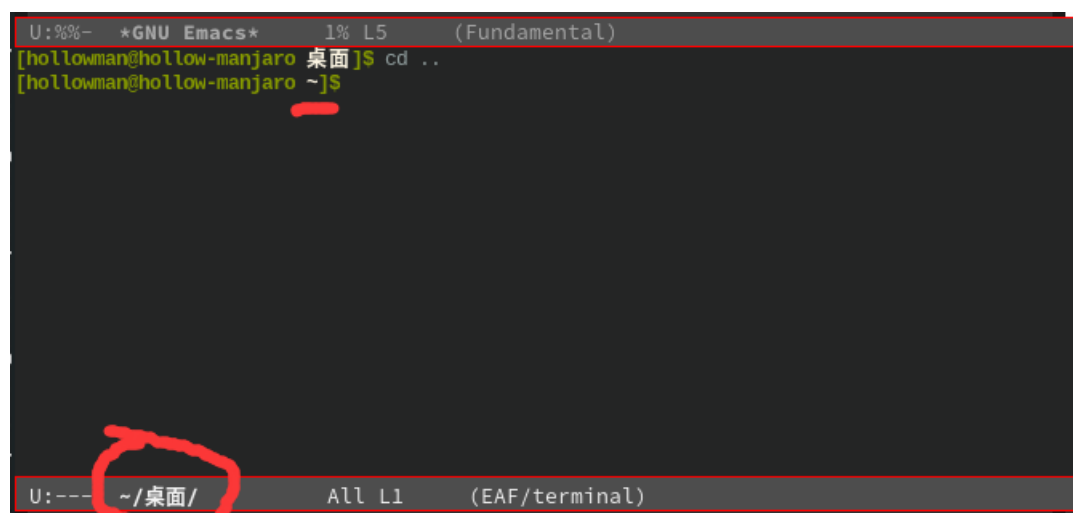
```
[hollowman@hollow-manjaro chemacs]$ exit
exit

U:--- ~/chemacs/ All L1 (EAF/terminal)
```

此任务将会通过修改 <https://github.com/manateelazycat/emacs-application-framework/blob/master/app/terminal/buffer.py> 等文件实现捕获信号，增加相关 emacs lisp 调用来实现。

D. 任务 4

该任务需要我们支持在 node-pty 中捕获更改目录信号，重命名终端缓冲区并实时更改 Emacs 默认目录变量：



```
U:%%~ *GNU Emacs* 1% L5 (Fundamental)
[hollowman@hollow-manjaro 桌面]$ cd ..
[hollowman@hollow-manjaro ~]$

U:--- ~/桌面/ All L1 (EAF/terminal)
```

与此任务与任务三比较类似，都可以通过增加相关 emacs lisp 调用来实现。

E. 任务 5-7

任务 5 需要我们使 Mindmap 支持从颜色列表中选择, 更改节点颜色。任务 6 需要我们使 Mindmap 支持多行插入 Dons 节点。该任务需要我们使 Mindmap 支持以正确的缩进级别导出 org。

为完成此任务, 我会尝试修改 <https://github.com/manateelazycat/emacs-application-framework/tree/master/app/mindmap> 这里的相关 js 脚本和 html 文件, 为 Mindmap 增加这些功能。

F. 任务 8

此任务需要为核心部分的 browser 增加 QTest::QTouchEventSequence 模拟触摸滚动事件的功能。我会根据相关 API: <https://doc.qt.io/qt-5/qtest-qtoucheventsequence.html>, 完成有关功能。可参考的修改点: <https://github.com/manateelazycat/emacs-application-framework/blob/master/core/browser.py#L164>

G. 任务 9

此任务需要使 browser 支持通过将密码存储在数据库中来自动填充密码。计划通过增加 JS, input="password" 来自动判断密码输入框, 并自动将密码使用 sqlite3 存储到数据库或者进行读取 (使用 <https://github.com/kripen/sql.js> 操作 sqlite3 数据库)。

H. 任务 10

此任务需要使 browser 支持不用鼠标进行网页的文本选择。此任务可参考 vimium 的实现方式: <https://superuser.com/questions/1236864/how-to-use-vimium-to-select-text-from-a-page> 即:

首先通过按 “/” 进入搜索模式, 输入好要复制的字符串前半部分, 然后按回车键返回普通模式后, 按 “v” 进入视图模式, 随后进行相关按键操作选择文本并进行复制操作。

此任务中文本的选择部分可以通过 JS 模拟鼠标的拖拽事件实现。

I. 任务 11

此任务需要使 browser 支持无鼠标浏览。现在 EAF 已经有了基础的 Vimium 功能, 这个任务是完善其 Vimium 功能使其能模拟真实的鼠标点击事件。此任务可以和任务 10 合并。

J. 任务 12

此任务需要使 Terminal 支持服务器准备就绪后刷新页面。这里可以设置相关机制, 在发出连接服务器命令之后, 进入等待状态, 判断服务器是否已经准备就绪, 完成之后再刷新页面。

K. 任务 13

此任务需要为 Terminal 修复连接建立错误：net::ERR_CONNECTION_REFUSED。项目具体细节将在工作时复现此 BUG 之后进行研究。

因为此 BUG 复现较为困难，如果工作过程中无法复现，我会选择“备 1”或者“备 2”任务作为我的替代方案。

L. 任务 14

此任务需要丰富 PDF 反转的颜色，参考 EAF 浏览器的 dark-mode 效果和实现：https://github.com/manateelazycat/emacs-application-framework/blob/master/core/js/dark_mode.js，使用 PyMuPdf 提供的接口<https://github.com/pymupdf/PyMuPDF/blob/master/docs/pixmap.rst>进行暗模式颜色算法的设计，使颜色和 emacs 当前主题的具体颜色联动起来。

M. 任务 15

此任务在任务 14 的基础上，要求排除暗模式下的图片。初步的纯反转模式下图片的排除已经被我实现。后续在任务 14 实现的基础上，此任务只需反运用任务 14 所提及的算法即可成功完成。

N. 备选任务 1

此任务需要使用 Qt 或 JavaScript 创建 Painter 应用。我将使用 Javascript，参考 Mindmap 的实现方式，参考这里：<https://github.com/yanhaijing/Painter>（MIT），并参考这里的效果：<https://github.com/chuntaro/epaint>进行实现。

O. 备选任务 2

此任务需要使用 JavaScript 重写 RSS Reader，我将根据这里的代码进行重构：<https://github.com/manateelazycat/emacs-application-framework/blob/master/app/rss-reader/buffer.py>

3. 时间安排

因为我暑期没有其它的安排，而且今年由于疫情原因，开学时间很有可能会推后，所以在 7 月份到 9 月份这 3 个月我可以全职工作（不限时间，每周至少 30 小时）。

我知道遇到问题时优先应该在互联网上搜索并参考有关资料，因而我只会和导师讨论自己无法克服的困难。我会每周通过电子邮件向导师发送一份“周期总结”，告知我到目前为止的工作进展和这周取得的进展。

以下是时间表：

A. 第 1 周 (7.1 - 5, 2020)

- 进一步了解 Emacs Lisp, DBus API, 查阅相关资料, 并且熟悉 EAF 代码库和其风格。

B. 第 2-6 周 (7.6 - 8.9, 2020) 共计 5 周

- 完成有关 Core 和 Mindmap 的任务 (所有都为低难度任务) (任务序号 1、2、5-7)。
- 完成有关 Terminal 的低难度部分任务 (任务序号 3、4)。
- 如果在此期间时间有剩余, 可以接着推进, 尽量提前完成有关 Terminal 中等难度的一两个任务。(如果无法复现任务 13 中的 Bug 则选择备选任务中的一个进行工作)

C. 第 7 周 (8.10 - 15, 2020)

- 准备中期报告。
- 为可能的拖延预留的赶进度时间。

D. 中期报告 (8.15, 2020)**交付成果**

- 所有达到任务目标的 Core、Mindmap 和 Terminal 低难度任务 (即所有低难度任务) 成果。(视实际情况可能会提前一两个项目的交付)。

E. 第 8-13 周 (8.17 - 24, 2020) 共计 5 周半

- 完成有关 Terminal 的中等难度部分任务 (任务序号 12、13) (如果无法复现任务 13 中的 Bug 则选择备选任务中的一个进行工作) (若前半段时间已经完成则免去这个任务)。
- 完成有关 Browser 的任务 (中等难度) (任务序号 8-11)。
- 完成有关 PDF 的任务 (中、高难度) (任务序号 14-15)。
- (可选) 如果时间有剩余, 我会挑战性地尝试将备选任务全部完成, 之后若还有时间剩余, 我会继续挑战 To-do List 其他任务。

F. 第 13-14 周 (9.25 - 9.30, 2020)

- 准备结项报告。
- 为可能的拖延预留的赶进度时间。

G. 结项(9.30, 2020)

交付成果

- 所有达到任务目标的 15 个任务成果。
- （可选）如果时间有剩余，我会挑战性地交付备选任务，之后若还有时间剩余，我会继续挑战并交付 To-do List 其他任务。

H. 未来 (10.1, 2020 -)

在成功完成此暑期 2020 项目后，我将继续为 EAF 做出贡献并尽我所能继续维护。如果可能的话，我会很高兴在未来的暑期开源活动中成为 GNU Emacs 社区的导师。