

## 开源实习报告

姓 名	蒋嵩林	学 校	兰州大学	年 级	大四
专 业	计算机科学与技术（基础理论班）	gitee ID	jiangsonglin2	积 分	150
实习时间	2022 年 1 月 10 日 —— 2022 年 7 月 9 日止				
实习 sig 组	sig-KIRAN-DESKTOP、Cloud Native、sig-OSCourse、sig-QA、sig-UKUI				

### 一.实习任务与成果

1. （sig-KIRAN-DESKTOP）适配 accountsservice 的 dbus 接口 30 分

issue 链接: <https://gitee.com/openeuler-competition/opensource-internship/issues/I4P81J>

成果 PR: <https://gitee.com/jiangsonglin2/kiran-cc-daemon/commit/b3251f66053c2dd7438ff34a0d9d93b485fea623>

2. （sig-OSCourse）为 Rust 工具 mdbook 实现 PDF 格式输出功能 50 分

issue 链接: <https://gitee.com/openeuler/opensource-intern/issues/I4QM4V>

成果 PR: <https://gitee.com/openeuler/opensource-intern/pulls/9> 、 <https://gitee.com/openeuler/opensource-intern/pulls/11> 、 <https://gitee.com/openeuler/opensource-intern/pulls/12> 、 <https://gitee.com/openeuler/opensource-intern/pulls/14>

3. （Cloud Native）云原生组件容器镜像镜像制作 - coredns 10 分

issue 链接: <https://gitee.com/openeuler/cloudnative/issues/I49Z2Y>

成果 PR: <https://gitee.com/openeuler/openeuler-docker-images/pulls/16>

4. （sig-QA）procinfo 软件包加固测试 10 分

issue 链接: <https://gitee.com/src-openeuler/procinfo/issues/I4TZ8G>

成果 PR: <https://gitee.com/openeuler/mugen/pulls/277>

5. （sig-UKUI）图形化备份工具的引入 20 分

issue 链接: <https://gitee.com/openeuler-competition/opensource-internship/issues/I4R5WP>

成果 PR: <https://gitee.com/openeuler/community/pulls/3305> 、 <https://gitee.com/src-openeuler/backintime/pulls/1> 、 <https://gitee.com/openeuler/community/pulls/3329>

6. (Cloud Native) 云原生组件容器镜像制作 - calico 30 分

issue 链接: <https://gitee.com/openeuler/cloudnative/issues/I49Z3A>

成果 PR: <https://gitee.com/openeuler/openeuler-docker-images/pulls/15> 、  
<https://gitee.com/openeuler/community/pulls/3685> 、 <https://gitee.com/src-openeuler/calico/pulls/1>

## 二. 实习总结

对于实习总结, 我认为最核心的是实习任务: 为 Rust 工具 mdbook 实现 PDF 格式输出功能, 其目标是为 mdbook 提供 PDF 输出功能, 在执行 mdbook build 的命令时, 根据配置项在 book 目录下生成 PDF 文件。

### ● 项目背景

mdbook 仅支持将书籍生成为 html 页面, 不支持对应的 pdf 输出。

### ● 项目需求

1. 完整功能实现, 包含一定的测试用例
2. 选择的依赖是通用的
3. 在 mac (m1)、Linux 大多数发行版和 windows 上都可以跑

### 开发环境要求:

1. 在 X86\_64 和 ARM64 架构下运行
2. Rust Edition 使用 2021

## 输出功能:

输出标准的 PDF 文件, 支持中文, 支持自定义 PDF 纸张方向、页面缩放比例、纸张宽度和高度、页面边距、生成的 PDF 页面范围、是否显示页眉和页脚以及自定义其格式。

## ● 技术实现方案

放弃了之前的 WebDriver 方案, 虽然 WebDriver 方案增加了对 Firefox 的支持, 但是由于一般 WebDriver 和浏览器版本之间是相互对应的, 用户需要手动查询浏览器版本, 下载对应的 WebDriver, 不如直接安装浏览器来的方便快捷。

而且使用 WebDriver 打印生成 PDF 文件时存在一个重大的缺陷, 遇到稍微大一点规模的文档时会由于页面崩溃导致生成失败 (Chrome 和 Firefox 对应的 WebDriver 都同时存在这个问题):

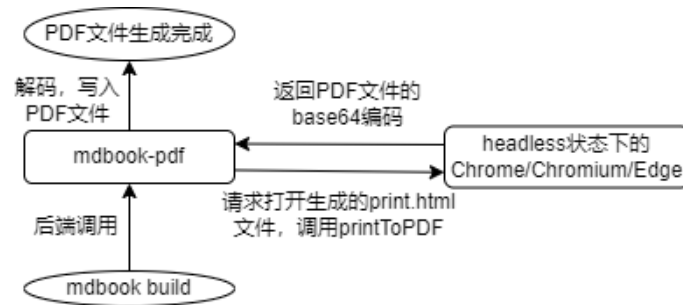
```
selenium.common.exceptions.WebDriverException: Message: unknown error: session
deleted because of page crash

from unknown error: cannot determine loading status

from tab crashed
```

此外 WebDriver 提供的可自定义打印 PDF 参数选项也相较而言不够多, 所以最终决定抛弃 WebDriver, 使用 Chrome 开发工具协议 直接和 headless 状态下基于 Chromium 的浏览器直接交互。

架构简图如下:



Rust Crate `headless chrome` 是 Chrome 开发工具团队维护的 Node JS 库 `Puppeteer` 的纯 Rust 实现，但是对于生成大的 PDF 文件的稳定性还不够，容易报超时错误，因而我通过 fork 该 Crate 仓库打补丁的方式，取消了超时限制：<https://github.com/HollowMan6/rust-headless-chrome/commit/0c5d707b28bdc577b5d386bc2e90d91b10143116>，同时增加了对调用 Edge 生成的支持：<https://github.com/HollowMan6/rust-headless-chrome/commit/c4517ef721f761e88a49f0005a729e792428fa0>

在补丁之后使用该 Crate 都能十分顺利地生成 PDF，且在同样的测试环境下也没遇到 WebDriver 下类似的崩溃错误。

目前该 `mdbook-pdf` 程序还未上架 `crates.io`，因为这里使用了 fork 仓库中的依赖，而并非源于 `crates.io`，所以离上架还遥遥无期，我已经提了相关的 issue 和 pr，希望能够在解决之后上游能 release 出新版本同步到 `crates.io`。

<https://github.com/atroche/rust-headless-chrome/issues/287>

<https://github.com/atroche/rust-headless-chrome/pull/288>

因而目前安装该程序的方式只能是通过仓库编译 `cargo build --release` 进行生成二进制文件使用，而不能直接使用 `cargo install` 的形式安装。对于 `x86_64` 的 macOS、Linux 和 Windows，以及 `arm64` 版本的 Linux 可以通过 <https://github.com/HollowMan6/mdbook-pdf/actions/runs/1737778711> 该工作流页面下方的 Artifacts 部分下载二进制文件直接使用。

**测试方案:**

<https://github.com/HollowMan6/mdbook-pdf/blob/main/.github/workflows/test.yml>

使用 GitHub Actions 进行 CI/CD，以 rust 相关书籍来进行测试，以能否成功生成 PDF 文档为判断标准。

<https://github.com/rust-lang/cargo/tree/master/src/doc>

<https://github.com/rust-lang-nursery/edition-guide>

<https://github.com/rust-embedded/book>

<https://github.com/rust-lang/mdBook/tree/master/guide>

[https://github.com/rust-lang/mdBook/tree/master/test\\_book](https://github.com/rust-lang/mdBook/tree/master/test_book)

<https://github.com/rust-lang/reference>

<https://github.com/rust-lang/rust-by-example>

<https://github.com/rust-lang/book>

<https://github.com/rust-lang/rustc-dev-guide>

<https://github.com/rust-lang/rust/tree/master/src/doc/rustdoc>

<https://github.com/rust-lang/nomicon>

目前所有上述书籍的 PDF 版本文档在 x86\_64 的 Linux, Windows, macOS 即使是无桌面环境下都可以正常生成，可点击此链接中的 Artifacts 部分查看测试 Workflow 生成的 PDF 文档。

## 开发资料

<https://doc.rust-lang.org/edition-guide/rust-2021/index.html>

[https://rust-lang.github.io/mdBook/for\\_developers/backends.html](https://rust-lang.github.io/mdBook/for_developers/backends.html)

<https://github.com/atroche/rust-headless-chrome>

## ● 成果

mdBook 支持使用自定义后端，当调用 `mdbook build` 命令时，假如书籍目录中的 `book.toml` 除了默认存在的 `[output.html]` 生成 html 网页端外，还存在 `[output.pdf]` 项，则会在系统 PATH 中调用 `mdbook-pdf`，同时该程序标准输入中传入相关书籍信息以及参数配置，具体细节请查看 mdBook 的说明文档 [Alternative Backends - mdBook Documentation](#)。

本程序使用 Rust 编写，根据上述原理基于 headless chrome 和 Chrome 开发工具协议，调用 Google Chrome / Chromium / Microsoft Edge 浏览器打开 mdBook 后端生成的网站中的 `print.html` 生成 PDF，由于是 headless 模式，运行时浏览器都在后台进行相应操作，无需图形界面，即使有图形界面也不会看到浏览器界面，无感知。

测试了一下 Firefox 对 Chrome 开发工具协议 打印生成 PDF 部分似乎不支持，Safari 浏览器对 Webdriver 打印生成 PDF 部分都不支持，因而遑论对 Chrome 开发工具协议 打印生成 PDF 部分的支持了，而对 Chrome / Chromium / Microsoft Edge 浏览器的最新版经测试都是支持的。

用户调用 `mdbook build` 命令时相关生成输出如下：

```
2022-01-22 03:21:43 [INFO] (mdbook::book): Book building has started
2022-01-22 03:21:43 [INFO] (mdbook::book): Running the html backend
2022-01-22 03:21:45 [INFO] (mdbook::book): Running the pdf backend
2022-01-22 03:21:45 [INFO] (mdbook::renderer): Invoking the "pdf" renderer
Generating PDF, please be patient...
PDF successfully generated at: /Users/runner/work/mdbook-pdf/mdbook-pdf/rust-by-example/book/pdf/output.pdf
```

因为 headless chrome 上游现在自动下载 Chromium 功能还不可用，因而现在某些情况下需要手动下载相关浏览器，即可正常使用。待后续如果上游修复了自动下载 Chromium 功能，将会同步更新，到时候用户将不用管任何东西直接运行即可。

- 在 Windows 10 及以上该程序无需安装任何额外软件即可正常生成 PDF，因为 Microsoft Edge 是 Windows 系统自带的浏览器。当然如果考虑到对没有自带安装 Edge 的老版本 Windows 的支持，在电脑上安装一个 Google Chrome 即可。

- 在 macOS 中需要下载并安装 Google Chrome。
- 在 Linux 中安装 Google Chrome / Chromium / Microsoft Edge 浏览器中的任意一个即

可，推荐安装 Chromium，该软件包在您的发行版中一般名称为 chromium 或 chromium-

browser，可参见 <https://pkgs.org/download/chromium> <https://pkgs.org/download/chromium-browser>

查看 Linux 发行版的支持情况（注意，在 Ubuntu 18.04 之后需要通过 snap 安装 chromium-browser）。

具体使用方法请参见 README：<https://gitee.com/jiangsonglin2/opensource-intern/blob/master/mdbook-pdf/README.md>

### 三. 个人成长/心得总结

很荣幸这次能够参与 openEuler 开源实习，贡献 openEuler 项目的建设。

这次活动中我接触到了许多之前从来没有接触过的东西，包括使用 C++ 调用 D-Bus、Glibmm、Rust 编程、Chrome 开发者工具远程调试协议、使用 mugen 编写测试。

最后，我要感谢项目进行期间各位来自 openEuler 贡献者的指导，是他们让我更多地了解了 openEuler 以及开源文化，对我进行了技术和细节上的指导，锻炼了我的团队协作和沟通能力，使我这次的 openEuler 开源实习参与收获丰厚。

因而，同学们，如果你对开源有足够的热情，推荐你参加 openEuler 开源实习活动！

### 四. 对参与开源社区贡献的理解与体会

参与开源贡献，于公而言，可以回报社区，提高知识共享水平，推动计算机产业的发展；于私而言，是一种低成本的提升自己技术水平的实战方式，可以在线和各式各样的技术大牛深度合作交流。因此，何乐而不为呢？