

项目申请书

岳泽龙

2020 年 6 月 16 日

组委会各位老师好，本人想要申请编号为 2021100，名为「为 openEuler 添加 Xfce 桌面环境并能够运行在树莓派 4B 上」的项目，下面将介绍我的技术方案以及时间规划。

1 技术方案

在我看来，该项目主要可以分为四个部分

1. 针对 openEuler 学习打包技术
2. 保证 Xorg Server、Xfwm、Xfce 等基础环境的运行
3. 保证 Xfce 相关软件的运行
4. 保证全新安装和镜像安装的可使用性

下面我将一一介绍这几点。

1.1 如何打包

在打包这方面，我曾经有过给 Arch Linux 的 AUR 仓库提交包的经验，分别是 [h2s](#) 以及 [ivideo](#)，还有现在已经转由官方维护的 [clash](#)。我自认为可以说是已经初步了解如何给 Arch Linux 打包了。同时也尝试过给 AOSC OS 打过包，但只是在本地尝试，并未提交到社区。因此，我觉得我具有一定的打包经验，也具有相应的学习能力和动力，对于使用 RPM 包管理器的 openEuler 来说，我可以查阅 [RPM Packaging Guide](#) 来学习相应的技巧，因此我觉得在打包方面应该是没什么问题。

在我看来，打包流程主要是以下几点

1. 找到包源码（如果是闭源包，则找到 binary）的来源，标清上游
2. 标出版本号

3. 填写 Hash 值
4. 分别找出构建包需要的依赖和包本身需要的依赖
5. 找到合适的编译选项 / 安装选项, 对包进行预处理 / 编译 / 定制
6. 指定 binary 和 lib, manpages 等安装的位置和权限

其中, 前三步, 甚至前四步, 都可以在已有软件的协助下进行完成, 而第五步, 则需要在软件官网或者软件文档中查询有关信息, 另外还要考虑直接编译还是交叉编译的问题, 较为复杂。第六步, 则需要按照标准执行, 对于某些写死了库位置的软件, 可能需要用 patchelf 来进行修改。

1.2 基础桌面

openEuler 一开始是作为服务器操作系统来进行开发的, 那么很可能是没有 Xorg 的相关 utils 的, 甚至 Xorg-server 也可能没有, 那么我们需要先把这些包打出来, 并检测 Xorg-server 和 Xorg-utils 的可用性。由于 Xfce4 暂时不支持 Wayland, 我们可以暂时不考虑对 Wayland 的相关软件打包。然后就是对 Xfwm, Xfce 进行打包了, 如果可用, 则可以输出图形画面并完成登录、注销、关机、重启等操作。

1.3 相关软件

上一步只完成了对基础桌面环境的打包, 但对于一个完整的桌面环境而言, 我们还需要网络管理器, 蓝牙设备管理器, 终端模拟器, 文件浏览器, 网络浏览器, 文本编辑器等软件, 还需要完整的中文字体, 因此我们还需要对这些软件进行打包, 对于 Xfce 有配套的, 我们可以直接选择 Xfce 配套的软件, 而对于没有配套的, 我们就需要进行一定的挑选, 例如选择体积较小的软件等。

1.4 安装验证

打包以及反复安装未经过测试的包, 都有一定概率污染 base 环境, 而导致依赖没有检测出来, 或者包有问题而无法得知。因此, 在打完所有包以后, 我们需要重新安装系统, 从头开始安装完所有包, 检测可用性, 并进行改进和修复。然后再将带桌面环境的系统打包成镜像, 再重新安装, 检测能否做到开箱即用。

2 时间规划

日期	安排
7.1 - 7.7	通过大量浏览 openEuler 现有包的 spec, 学习 RPM 打包规范, 并尝试给 openEuler 打一些源里暂时没有的包进行练手
7.8 - 7.14	根据 openEuler 的实际需求, 首先查看 X11 相关的工具以及软件包是否齐全, 如果 Xfwm 和 Xfce 有相关依赖不在软件源中, 则进行补充, 尝试在本周内能打出可运行的 Xfwm
7.15 - 7.21	在本周内尝试打出 Xfce, 并添加默认的字体的配置文件, 如果所需的中文 / 英文 / 等宽字体在软件源中不存在, 则进行补充。并检测登录 / 注销 / 重启 / 关机等功能能否正常运行
7.28 - 8.3	在本周内为已有的 Xfce 桌面环境进行一部分功能的添加, 例如网络功能, 检查 NetworkManager 的状态, 并打出 Xfce 的网络管理器, 蓝牙管理器等, 并进行蓝牙以及网络图形界面管理功能的验证
第一阶段	第一阶段的主要任务是学习 RPM 打包规范, 补充 Xfce 依赖的相关软件包, 并打出一个具有基本功能, 可以正常使用的 Xfce 桌面环境
8.4 - 8.10	在本周内打完大部分的 Xfce 配套软件包, 诸如终端模拟器, 文件管理器, 浏览器等等, 并逐一测试功能, 对于 Xfce 团队没有额外开发, 但是工作 / 生活中会用到的软件, 进行挑选, 打包, 并在将来撰写的文档中进行推荐
8.11 - 8.17	完成中期检查报告, 对项目完成度进行评估, 将没有完成的部分进行整理, 并根据实际进度进行后续的工作安排, 对前一段的工作进行总结, 不足的地方进行反思
8.18 - 8.24	将还未打出的包进行补充, 尽量在本周内完成所有相关包的打包, 并对所有包进行记录整理, 功能测试, 等等
8.25 - 8.31	将已有的系统清空, 在干净的存储卡上全新安装, 记录安装过程, 编写文档。将安装过程中遗漏的依赖, 软件包等记录下来。对安装的系统再次进行功能测试
第二阶段	第二阶段的主要任务是将 Xfce 相关所有的软件包打完, 并撰写一部分文档, 验证可重复安装性, 为结束该任务做准备
9.1 - 9.7	对前一周记录的依赖, 软件包进行添加, 完善文档, 并按照新打出的包重新进行全新安装, 如有问题, 则进一步解决
9.8 - 9.14	将桌面环境以及 base 环境以镜像的形式进行打包, 测试镜像, 修复镜像, 并根据实际情况, 清理非必要包, 对镜像体积进行精简
9.15 - 9.21	按照之前的记录, 完善文档, 介绍 Xfce 的使用, 推荐一部分非必要软件包。进行最后的测试和总结, 修复能发现的漏洞, 梳理提交
9.22 - 9.30	对本次活动进行总结, 撰写总结报告, 按照导师和项目组的要求完成评价, 总结等等, 并对自己整个阶段的表现和收获进行整理, 发布在博客上, 完成整个活动的参与阶段
第三阶段	第三阶段的主要任务是撰写完文档, 将安装过程及使用过程中可能遇到的问题尽可能找到, 并进行修复, 尽力提高项目的完成度, 结束项目