AOSCC 2025 | SJTUG

deepin 桌面环境在发行版移植中的问题

Cryolitia, 项泽龙

2025-07-25

deepin 社区

目录

1. 上游依赖项补丁有关的耦合问题

deepin 系统仓库中的第三方库 ABI/API 与上游不兼容，导致 DDE 在其他发行版上崩溃

1. 上游依赖项补丁有关的耦合问题

1.0.1 来源

- DDE 项目的研发过程中直接在 deepin 系统仓库中对第三方库进行了破坏 ABI/API 兼容性的修改
- 出于某些需要，DDE 项目依赖了仅在 deepin 环境下需要的闭源模块

1.0.2 解决方案

要求 DDE 项目必须兼容原始上游项目

1.0.3 RFC（发起中）

rfc: patch maintenance convention

2. 大量 DTK 项目 CMake 文件中的版本号存在人为性错误

CMake 版本号与实际版本号不一致，版本号需在编译时传入

2.1.1 CMake

```cmake
set (DTK_VERSION "5.6.12" CACHE STRING "define project version")
project (DtkCore
  VERSION ${DTK_VERSION}
  DESCRIPTION "DTK Core module"
  LANGUAGES CXX C
)

if("${PROJECT_VERSION_MAJOR}" STREQUAL "5")
    set(QT_VERSION_MAJOR "5")
elseif("${PROJECT_VERSION_MAJOR}" STREQUAL "6")
    set(QT_VERSION_MAJOR "6")
    set(DTK_VERSION_MAJOR "6")
else()
    message(SEND_ERROR "not support Prject Version ${PROJECT_VERSION}.")
endif()
```

2.1.2 Arch Linux PKGBUILD

```bash
build() {
  cd dtk6core
  cmake . -GNinja \
      -DBUILD_DOCS=ON \
      -DBUILD_TESTING=ON \
      -DCMAKE_INSTALL_LIBDIR=lib \
      -DCMAKE_INSTALL_LIBEXECDIR=lib \
      -DCMAKE_INSTALL_PREFIX=/usr \
      -DDTK_VERSION=$pkgver
  ninja
}
```

2.2.1 原因

- 平行项目形如 `dtkcore` 和 `dtk6core`
- 研发认为版本发布时仅修改一处 `debian/changelog` 中的版本号即可，更多修改是不必要的
- `dtkcore` 和 `dtk6core` 之间的变更被魔法同步，两个仓库中的 cmake 文件均同时包含 Qt5 和 Qt6 的版本
- CMake 在编译时根据传入的版本号确定 Qt 版本
- 由于某大型软件仓库已经在主线接受了在编译时传入版本号的魔法操作，导致这个行为被错误认为是非常合理的

2.2.2 解决方案

- 设计自动化发版流程，用更懒的操作防止研发使用先前偷懒的错误方式
- 所有项目对 `debian` 目录的更改必须经过 deepin 系统组审核

3. DDE 与 deepin 配置文件混杂

deepin-desktop-base
deepin-desktop-schemas
deepin-osconfig
default-settings

3. DDE 与 deepin 配置文件混杂

3.0.1 现状

- deepin/dde 配置梳理 · linuxdeepin · Discussion #11769
- `deepin-desktop-base` 看起来像 DDE 相关，但打包安装后 `os-release` 神奇的变成了 deepin
- `deepin-osconfig` 看起来像 deepin 系统相关，但不打包安装会导致 DDE 无法正常使用。

3.0.2 解决方案

- `dde-settings-default`
- `deepin-settings-default`

4. Qt 私有头文件的使用

4.0.1 现状

- DDE 项目出于实际需要，使用了 Qt 的私有头文件，而许多发行版没有提供这些头文件
- Qt 私有库部分的 ABI 不稳定，可能导致运行时找不到符号而崩溃

4.0.2 解决方案

- 直接向其他发行版推动 Qt 私有头文件的打包
  - Debian 中已经包含大量形如 `qt*-private-dev` 的包
  - 先进的构建系统 Nix 可以直接通过 `${qt6Packages.qtbase.src}` 引用
- 暂时在其他发行版上注释或回退使用不稳定符号的相关代码

5. 其他问题

5. 其他问题

- 部分项目安装过程中假设了特定的 `libexecdir` 目录，已经更改为允许在构建过程中通过参数传入
- 部分项目安装过程中文件的权限位设置不正确，例如，可执行文件未设置 `x` 权限位
- 大量 DDE 项目缺乏供发行版参考的清晰有意义的自然语言描述
- 部分已经废弃不再使用的项目仍作为依赖被其他软件引用，仓库也没有被标记废弃或存档

AOSC 构建系统中严格的品质保证（QA）检查帮助我们发现了大量未曾察觉的上述错误，在此特向 AOSC 表示感谢。

感谢！