

Building Qt from Source on Linux

Liyuan Liu Nan Li Pengtao Zou Xinghui Wang

North China Institute of Aerospace Engineering, Langfang, Hebei, 065000, China

Abstract

Qt is a development framework that can be used to create GUI programs or non-GUI programs. Applications developed with Qt can run on a variety of platforms, including desktop and embedded systems. The open source version of Qt 6 no longer provides an offline installation package, and you need to compile Qt yourself. In this paper, the Qt 6 source code is compiled using the Linux system and compilation technology. First, the importance of building Qt from source in Linux is analyzed. Secondly, the complete construction method is described. Finally, use Qt to develop a GUI program and verify the correctness of the build. It is hoped that through this practice, it can help improve the stability and security of the application products developed by Qt, and provide effective reference and reference for relevant personnel.

Keywords

Linux; Qt; source

在 Linux 上从源代码构建 Qt

刘立媛 李楠 邹澎涛 王兴会

北华航天工业学院, 中国 · 河北 廊坊 065000

摘要

Qt是一个开发框架,可以用来创建GUI程序或非GUI程序。用Qt开发的程序可以运行在包括桌面和嵌入式系统的不同平台。Qt 6开源版本已不提供离线安装包,需要自行编译Qt。论文综合运用Linux系统和编译技术,完成对Qt 6源代码的编译。首先,分析了在Linux中从源代码构建Qt的重要性。其次,阐述了完整的构建方法。最后,使用Qt开发GUI程序并验证构建的正确性。希望通过这次实践,能对提高Qt开发的应用产品的稳定性和安全性起到帮助作用,并为相关人员提供有效的借鉴和参考。

关键词

Linux; Qt; 源代码

1 引言

近些年来 Qt 的版本不断更新,并且不再提供开源版本的离线安装包。通过在 Linux 中从源代码构建 Qt,可以在离线方式下配置 Qt,为减少网络等待和保障信息安全提供重要的技术支持。论文将从源代码构建 Qt 的重要性、构建过程和使用 Qt 编写简单的 GUI 测试程序并运行三方面进行阐述。

2 从源代码构建 Qt 的重要性

虽然 Qt 的在线安装方式操作简单,对比在 Linux 中从源代码构建 Qt 则是一项烦琐的任务,需要熟悉编译过程和技术,但是从源代码构建可以为程序员提供更大的自由度和定制性,帮助程序员在遇到特殊情况时更好地满足自己的需求。从源代码构建带来的好处主要体现在离线安装和编译源码上。

【作者简介】刘立媛(1980-),女,中国河北张家口人,本科,讲师,从事计算机应用研究。

2.1 离线安装方式

①安全性:离线方式安装可以确保安装过程不被外部因素干扰,保证了系统的安全性。

②完整性:离线方式安装不会因为网络问题、依赖项缺失而导致安装失败或安装不完全。

③灵活性:离线方式安装可以使用户在任何时间、任何地点都能够进行自主选择安装,而不需要依赖互联网连接。

④节约资源:离线方式安装可以使用户节省大量的时间和精力,避免在线安装时可能遇到的下载速度慢、中断等问题。

总之,离线安装可以提高软件的安全性、完整性、灵活性及节约资源,对于一些需要在不稳定的网络环境中,或者是在没有网络的环境中进行工作的人来说是非常有用的。

2.2 编译源码方式

①更灵活的选项:源码编译提供了一些配置选项,可以在源代码级别上添加或删除模块、启用或禁用特性等。例

如构建 Qt 过程中 `configure` 的使用。

②使用最新版本：源码编译提供了可以让用户使用的 Qt 是最新的版本的途径，使用最新的版本可以获得最新功能和 bug 修复。例如，目前最新版本的 Qt 6.6，它为 Qt Quick 引入了一个新的布局元素，即 `LayoutItemProxy`，通过使用它，界面设计人员可以创建不同的布局，并在不同布局之间切换；Qt Graphs 模块将旧的基于 OpenGL 的 Qt DataVisualization 和 Qt Charts 模块的功能组合到一个单一的、现代的 Qt 6 框架中，借助 Qt Graphs，应用程序可以对大量快速变化的数据进行可视化，例如条形图、散点图和曲面图；扩展了屏幕捕获这个功能，允许应用程序捕获单个窗口^[1]。关于 Qt 6.6 更多新功能的详情可参考官方公告。

③适合特定环境：源码编译可以让用户根据自己的特定环境需求进行定制。例如 Qt 经常被应用于嵌入式系统的程序开发。

④确保程序安全：通过检查源码中的漏洞和安全隐患，可以及时修复并防止潜在的安全风险。这意味着通过编译 Qt 源码，有助于提高开发环境的安全性，从而也有助于开发的程序的安全性。

3 构建过程

论文阐述的构建过程使用的 Linux 为 Ubuntu 22.04，Qt 为 Qt 6.6。遇到的很多官方文档中未详细描述的细节问题，经不断查找资料、分析论证、多次实验后予以解决，在这里不再一一赘述。在整个实验过程中，都使用了 root 账户登录以方便操作，涉及的相关主要目录可以先创建好，目录结构如图 1 所示。

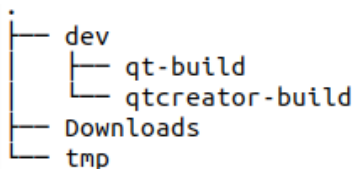


图 1 目录结构

3.1 构建准备

要从 Qt 的源代码构建 Qt，需要注意的是要在系统中安装一些相关库或工具，其具体名称可参考官方文档^[2]。

3.2 获取源代码

下载 `qt-everywhere-src-6.6.0.tar.xz` 至 Downloads 目录中，将其解压缩到 tmp 目录，命令如下：

```

root@localhost:~# cd ~/tmp
root@localhost:~/tmp# tar xf ~/Downloads/qt-everywhere-
src-6.6.0.tar.xz
  
```

3.3 安装构建环境

构建 Qt 需要一个包含工具、编译器的构建环境。

3.3.1 构建工具

需要使用的工具有 CMake、Ninja 和 Python，安装的版

本如下：

```

root@localhost:~# cmake --version

cmake version 3.22.1

CMake suite maintained and supported by Kitware
(kitware.com/cmake).
  
```

```

root@localhost:~# ninja --version

1.11.1
  
```

```

root@localhost:~# python3 --version

Python 3.10.12
  
```

3.3.2 编译器

一些相关版本信息如下：

```

root@localhost:~# cat /proc/version

Linux version 6.2.0-36-generic (buildd@lcy02-amd64-050)
(x86_64-linux-gnu-gcc-11 (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0,
GNU ld (GNU Binutils for Ubuntu) 2.38) #37~22.04.1-Ubuntu
SMP PREEMPT_DYNAMIC Mon Oct 9 15:34:04 UTC 2
  
```

3.4 构建 Qt 库和工具

若要配置生成，要创建一个生成目录，并在此目录中运行 `configure` 脚本。

```

root@localhost:~# cd ~/dev/qt-build

root@localhost:~/dev/qt-build# ~/tmp/qt-everywhere-
src-6.6.0/configure
  
```

默认情况下，Qt 配置安装在 `/usr/local/Qt-6.6.0` 目录中。

继续构建库和工具：

```

root@localhost:~/dev/qt-build# cmake --build . --parallel

生成后，要在适当的位置安装库和工具：

root@localhost:~/dev/qt-build# cmake --install .
  
```

4 使用 Qt

上述步骤顺利完成后，就可以开始使用 Qt 构建应用程序了。此时可以通过命令行方式或借助 Qt Creator 的方式来验证。

4.1 命令行方式

将 Qt 工具添加到默认 PATH 中。具体操作如下：

```

在 .profile 中添加以下行：

PATH=/opt/Qt/6.6.0/gcc_64/bin:$PATH

export PATH
  
```

环境变量生效后，编写测试程序 `HelloWorld`，在终端经过直接使用 `qmake` 等一系列操作后，就可以生成能够运行的 `HelloWord`，运行效果如图 2 所示。

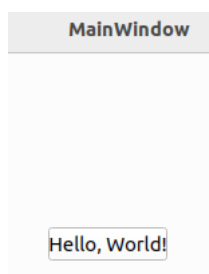


图 2 测试程序运行

4.2 Qt Creator 方式

Qt Creator 是一个集成开发环境，虽然 Qt Creator 的安装不是必需的，但是在实际编程中使用 Qt Creator 可以更加直观地新建项目和调试项目，论文给出 Qt Creator 的源代码安装方法，有需要的用户可以参考^[1]。

①安装构建工具。

在终端输入以下命令：

```
root@localhost:~/dev/qtcreator-build# apt install build-essential cmake ninja-build git
```

②安装 sdk。

在终端输入以下命令：

```
root@localhost:~/dev/qtcreator-build# apt install qt6-base-dev qt6-base-private-dev qt6-declarative-dev qt6-declarative-private-dev qt6-tools-dev qt6-tools-private-dev qt6-scxml-dev qt6-documentation-tools libqt6core5compat6-dev qt6-tools-dev-tools qt6-l10n-tools qt6-shader-baker libqt6shadertools6-dev qt6-quick3d-dev qt6-quick3d-dev-tools libqt6svg6-dev libqt6quicktimeline6-dev libqt6serialport6-dev
```

③安装 LLVM/Clang 依赖。

在终端输入以下命令：

```
root@localhost:~/dev/qtcreator-build# apt install clang-17 clangd-17 libclang-17-dev
```

④安装 Qt Creator 依赖。

在终端输入以下命令：

```
root@localhost:~/dev/qtcreator-build# apt install libgl1-mesa-dev libvulkan-dev libxcb-xinput-dev libxcb-xinerama0-dev libxcb-common-dev libxcb-common-x11-dev libxcb-image0 libxcb-keysyms1 libxcb-render-util0 libxcb-xkb1 libxcb-randr0 libxcb-icccm4
```

⑤获取 Qt Creator 源代码。

在终端输入以下命令：

```
root@localhost:~/dev/qtcreator-build# git clone --recursive https://code.qt.io/qt-creator/qt-creator.git repo
```

⑥构建 Qt Creator。

在终端输入以下命令：

```
root@localhost:~/dev/qtcreator-build# cmake -G Ninja -S repo -B build -D CMAKE_BUILD_TYPE=Release
root@localhost:~/dev/qtcreator-build# cmake --build build
root@localhost:~/dev/qtcreator-build# cmake --install build/ --prefix qtcreator-install
```

至此，若成功构建，即可使用以下命令启动 Qt Creator。

```
root@localhost:~/dev/qtcreator-build# ./qtcreator-install/bin/qtcreator
```

经过一些配置后就可以在 Qt Creator 中新建和调试 HelloWorld 项目了，运行结果也可以参考图 2。

实践证明文中所述源代码构建 Qt 的方法切实可行。

5 结语

论文阐述了源码构建 Qt 的重要性，经过长时间的实践，不断调试后，给出了构建 Qt 的可行方法。希望能有助于其他用户学习和使用 Qt，在此也对在论文撰写过程中提供学术帮助和支持的人和机构表示感谢。

参考文献

- [1] The Qt Company. <https://www.qt.io/blog/qt-6.6-released>.
- [2] The Qt Company. <https://doc.qt.io/qt-6/linux-building.html>.
- [3] Qt Wiki https://wiki.qt.io/Building_Qt_Creator_from_Git_on_Ubuntu_22.04.