**Qt开发笔记**

23090032047 计算机类1班 于景一

1. 开发思路

考虑其功能的实现方法；考虑如何正确设置开发环境；考虑是否要实现跨平台的特性，如何实现，如何封包。

* 1. 实现方法

1. 计算模块：使用基于逆波兰表达式的双栈法进行表达式计算。

见：源码/CalMethodByJS.cpp

文本

描述已自动生成

图1.1-1 表达式计算程序的一种实现

1. Signals and Slots：

**对此二者的理解如下：**

Signal是当某个对象的状态发生变化时发出的一种通知。Qt的控件有许多预定义的信号，但我们也可以通过继承控件来添加自己的信号。

Slot是响应某个信号而调用的一种函数。Qt的控件有许多预定义的插槽，但我们也可以通过继承控件来添加自己的插槽。

信号和插槽是松耦合的：发出信号的对象不知道也不关心有哪些插槽接收了信号。Qt的信号和插槽机制保证了如果我们将一个信号和一个插槽连接起来，那么在适当的时候，插槽会被信号的参数调用。

**如何链接此两者：**

连接信号和插槽：我们可以使用QObject::connect函数来连接信号和插槽。我们可以使用基于函数指针的语法，也可以使用基于字符串的语法。基于函数指针的语法更加简洁和类型安全，但是基于字符串的语法更加灵活和动态。

**参看：源码/JSCalculator.cpp: Line 170..199**

文本

描述已自动生成

图1.1-2 在计算器程序中的应用

* 1. 设置开发环境

① 下载并安装Qt

在<https://download.qt.io/official_releases/online_installers/>获取到正确Online Installer，在无商用License的时候会默认下载开源的版本。

对于Windows选择qt-unified-windows-x64-online.exe

对于Linux选择qt-unified-linux-x64-online.run

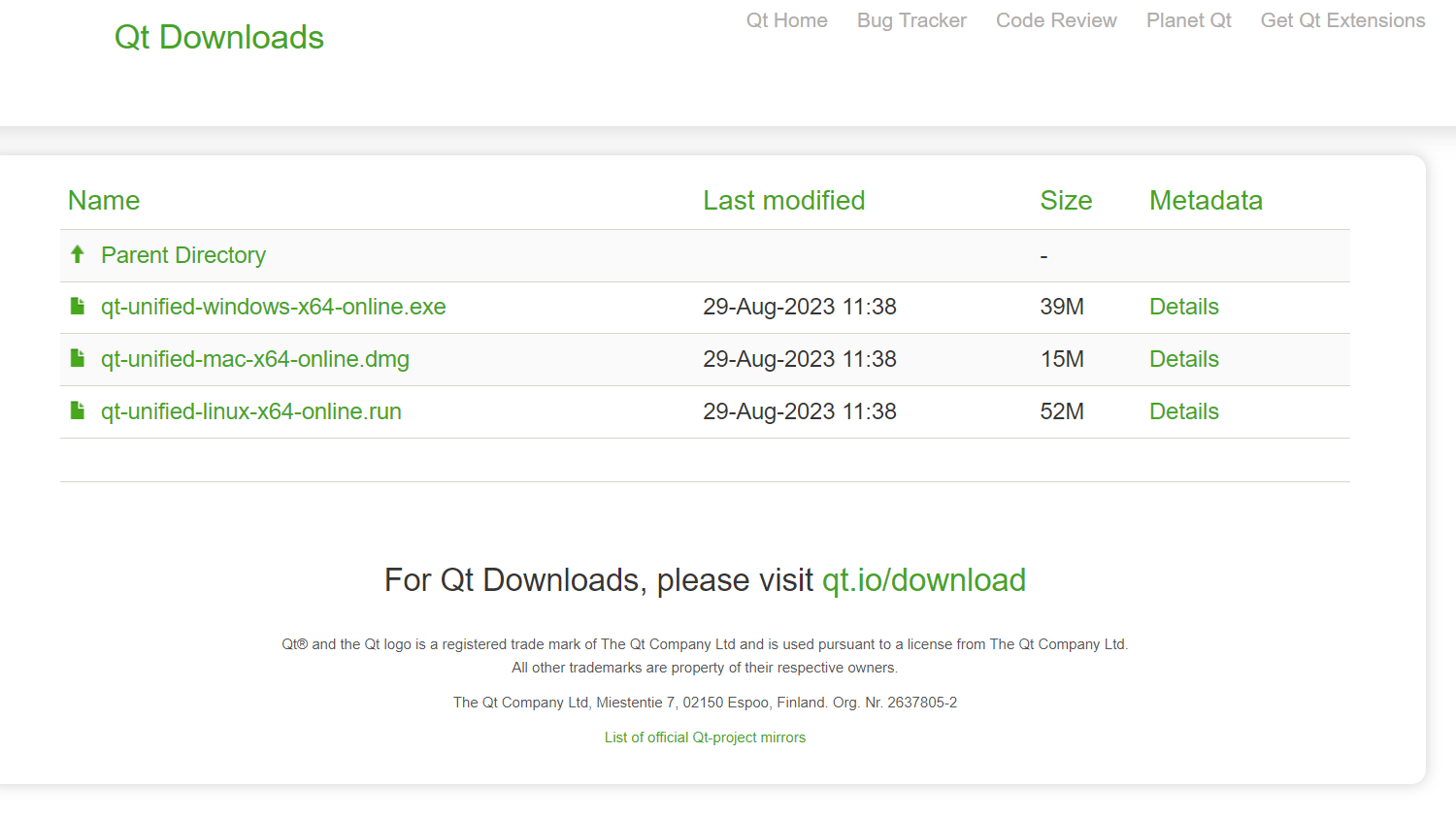


图1.2-1 下载Qt

**② 使用WSL2建立Windows下的Ubuntu子系统**

考虑到Windows11带有子系统的功能，可以在Windows运行之上实现Linux子系统的运行，做到无缝衔接，便于开发，且内存占用率低。在这里提供简单的实现方法：

文本

中度可信度描述已自动生成

图1.2-2 正确配置后的WSL列表

简单给出一个设置方法：

wsl --update

wsl --set-default-version 2

wsl --export Ubuntu-22.04 \*\*\*\*\ubuntu22.04.tar

wsl --unregister Ubuntu-22.04

wsl --import Ubuntu-22.04 \*\*\*\*\UbuntuWSL \*\*\*\*\ubuntu22.04.tar --version 2

ubuntu2204.exe config --default-user USER\_YOU\_SET\_BEFORE

sudo passwd root

对于一些基本的带GUI窗口，WSL2默认可以实现（基于Wayland）。

配置系统基本的程序见微软的文档：

<https://learn.microsoft.com/zh-cn/windows/wsl/tutorials/gui-apps>

下面给出正确运行WSL2的证明。

图形用户界面, 文本

描述已自动生成

图1.2-3 在WSL2的Ubuntu 22.04 LTS运行Linux QQ

* 1. 跨平台的实现

Qt具有显而易见的跨平台性，然而在程序也应注意这一点。

如果我要仅在Windows下调用WinAPI（引用windows.h），而在其他平台用其他的方法来替换，则参考：

#ifdef CONDITION\_1

#include <Windows.h>

#else

qDebug() << "For other platforms, UI Effect is defaultly disabled.";

#endif

在本项目中，请看：源代码/JSCalculator.cpp 及其他。

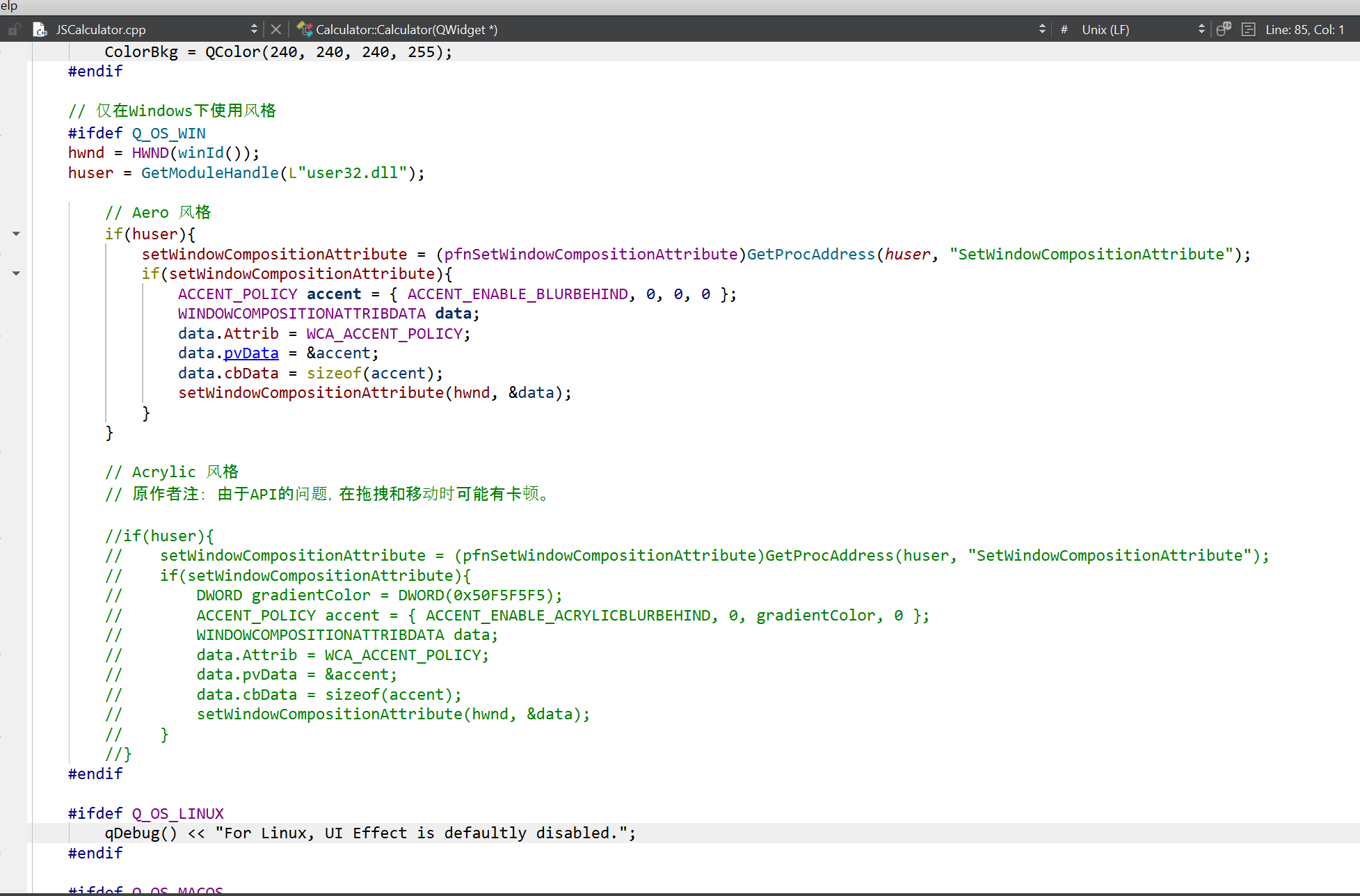


图1.3 Qt对应平台的设置

* 1. 如何封包

在Windows考虑：windeployqt

在Linux考虑linuxdeployqt或Go\_AppImage等

在此不加赘述。

图形用户界面, 应用程序

描述已自动生成

图1.4 在Windows下封包成功结果

1. 实验日志
   1. 编写程序源码

文本

描述已自动生成

图2.1 源码成果

**说明：**

1. 本程序在 Qt6 以下版本可能无法很好地工作。

2. 我也原创了一个通过逆波兰表达式（后缀表达式）处理复杂算式 （可以是含括号的表达式）的算法。请看源码./CalMethodByJS.cpp

3. 通过 #ifdef Q\_OS\_WIN ... #endif 实现平台判断，在Windows下调用WinAPI实现模糊效果

4. 自定义程序的外观，参考了 Acrylic / Aero 设计风格，同时实现了程序宽度增加后，显示右侧菜单的类Windows设计。这部分的实现请从 JSCalculator 第一部分开始看。

这里引用了部分代码，具体参见： @Linloir (GitHub) 的设计 （Under GPL-2.0 License）

原作者使用了一些已Deprecated的函数或方法，我进行了修复，删除掉很多unused参数，如event, expStr等。

* 1. 移植到Ubuntu下构建

有了1.3的基础，这个很简单解决。

您可以发现，在Linux下，WindowCompositionAttribute.h默认被禁用。

图形用户界面, 文本, 应用程序

描述已自动生成

图2.2-1 在Linux下WindowCompositionAttribute默认禁用

构建出一个可运行的二进制程序JSCalculator：

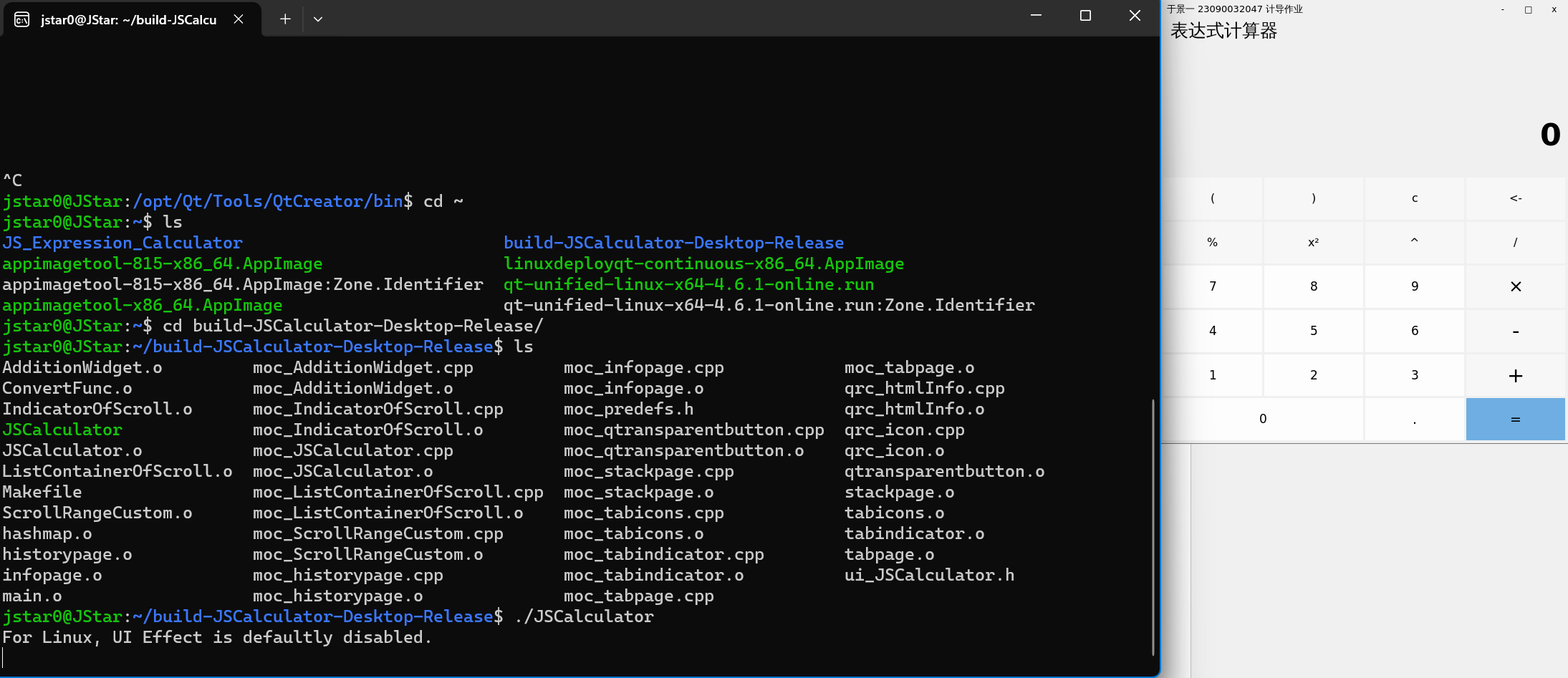


图2.2-2 在Linux下构建

然后进行跨平台处理即可。

1. 正确运行的截图（脱离QtCreator）
   1. 在Windows下

表格

描述已自动生成

* 1. 在Linux（Ubuntu 22.04 LTS）下

图形用户界面, 文本

描述已自动生成

**23090032047 于景一**

**2023/11/26**