

# Qt Shell 项目开发文档

计算机与信息科学学院 软件工程专业 2021 级 范榆康、唐雨

## 1. 需求分析

### 1.1. 背景

Shell 是操作系统中一个重要的组件，负责命令的解析和执行。为了在图形界面环境中提供类似的功能，本项目旨在使用 Qt6 和 QML 实现一个图形化的 Linux shell 程序。

### 1.2. 目标

创建一个基于 Qt 6 和 QML 的图形化 Linux shell 程序；  
实现良好的代码体系结构，使用 CMake 进行构建，Git 进行项目管理；  
提供基本的 Linux shell 命令处理能力、Shell 脚本运行能力，IO 重定向功能；  
完善的软件配套开发文档，

### 1.3. 用户需求

目标用户：Unix/Linux 用户；计算机科学学生。

### 1.4. 功能需求

命令输入：提供一个输入框供用户输入命令；  
命令输出：在一个显示区域中展示命令的输出结果；  
环境变量管理：支持设置和获取环境变量；  
脚本执行：可以执行自定义的 Shell 脚本文件。

### 1.5. 非功能性需求

性能需求：程序应能够快速响应用户输入并显示命令执行结果；程序能够高效地处理并行执行的多个命令；

兼容性需求：支持各种版本的 Linux、Unix 系统；

可维护性需求：代码具有高度的封装性，使用 CMake 进行项目构建，具有优

秀的文件组织架构。

## 2. 架构设计

### 2.1. 用例设计

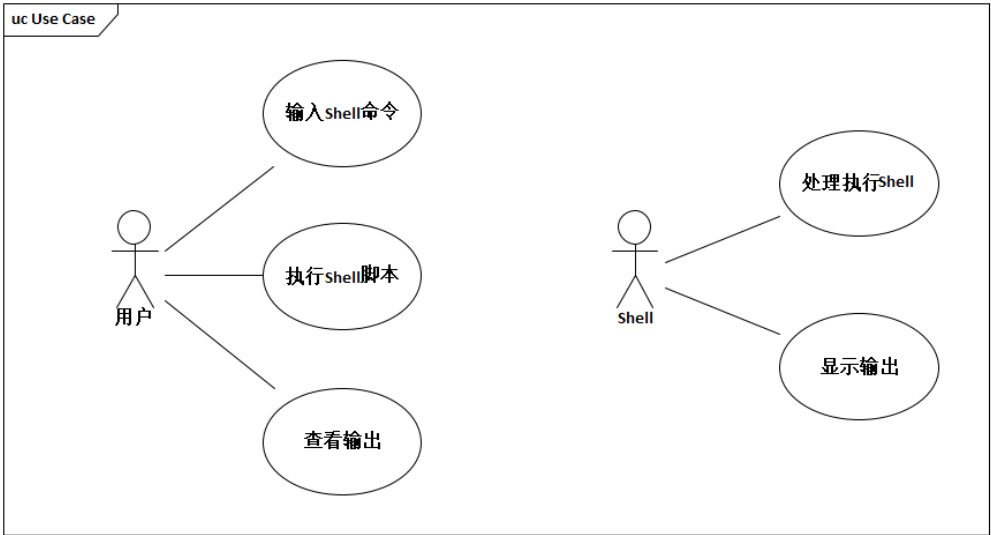


图 1 Use Case

如图 1，主要用例为“用户”，用户具有输入 Shell 命令、执行 Shell 脚本、查看输出功能；Shell 对用户的操作进行处理，执行 Shell 和 I/O 重定向管道进行输出。

### 2.2. 软件层次结构

界面层（UI）：使用 QML 实现，包括命令输入框、输出显示区

业务逻辑层（Business Logic）：处理用户输入的命令，调用相应的处理函数并返回结果

数据层（Data Layer）：管理 Shell 环境变量和命令执行的具体实现

### 2.3. 模块划分

**Controller**：负责接收用户输入的命令，调用业务逻辑层进行处理，并将结果返回给界面层

**Shell**：具体处理用户输入的命令，调用数据层执行命令并返回结果

### 2.4. 类图

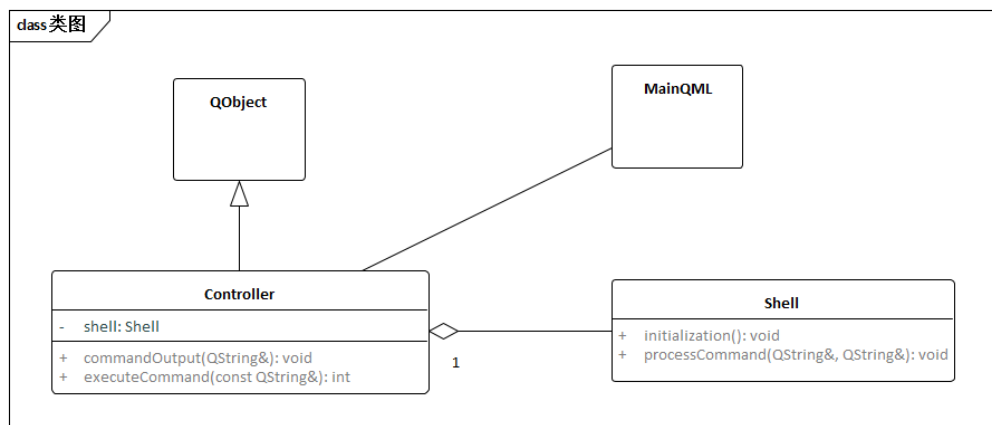


图 2 类图

如图 2，Controller 继承 Qt 的 QObject 基类，具有一个 Shell 类实例

## 2.5. 流程图

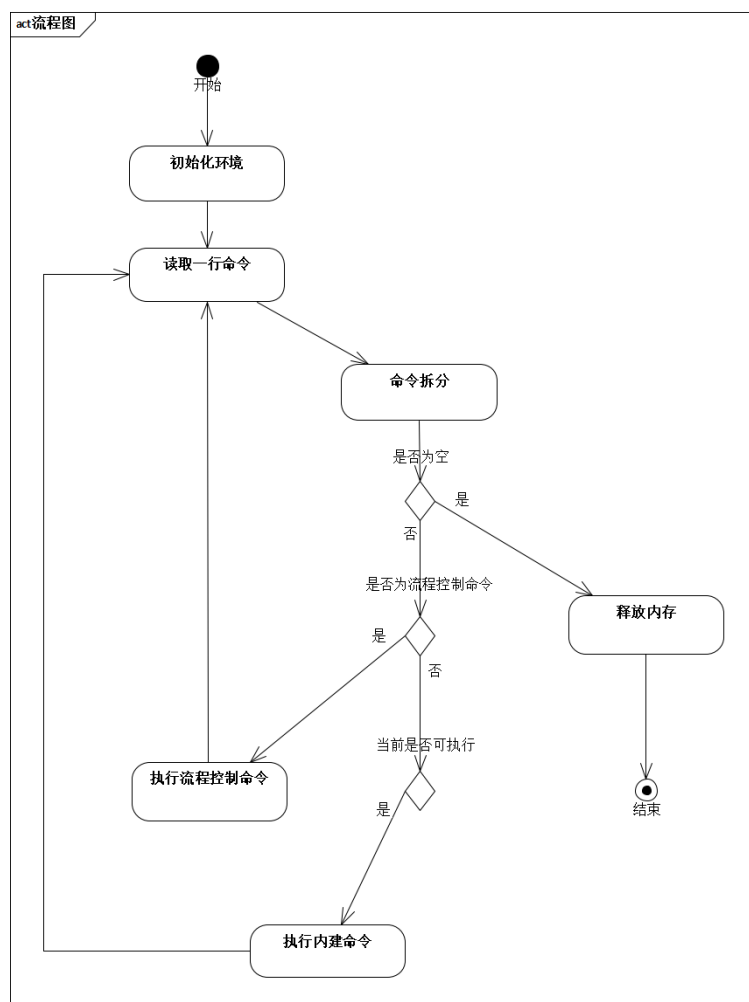


图 3 流程图

如图 3，为系统的流程图。开始运行软件是，mian 主函数进行初始化并注册 QML 界面、初始化 Controller 类。

**Controller** 构造时初始化环境，开始命令的识别。

**Shell** 类每次识别一行命令，将命令拆分；如果为空则关闭结束，否则检测是否为流程控制命令（**if**、**else**、**fi**）。是则进行流程控制命令，返回继续读取下一行命令；否则检测是否可以执行，开始执行内建命令，完成后继续读取下一行。