

基于 Qt 的 Android 应用程序 C/C++ 开发方法与实践^①

闫锋欣, 牛子杰, 杜烁炜, 潘天丽

(西北农林科技大学 机械与电子工程学院, 杨凌 712100)

通讯作者: 牛子杰, E-mail: 328920539@qq.com

摘要: Android Java 应用程序 (APP) 运行时需依赖 Dalvik 虚拟机环境, 从而影响 APP 的性能表现. C/C++ 程序具有极好的硬件适应能力, 但不能直接开发 Android APP. 针对该问题, 利用 Dalvik 虚拟机支持 JNI (Java Native Interface) 调用的特点, 设计了基于 Qt 的 Android APP C/C++ 开发方法框架体系, 给出了框架的详细工作机制和解决方案, 系统总结了利用 C/C++ 开发 Android APP 的主要过程和步骤. 最后, 以 Android 版机械 CAD 导教 APP 的开发工作为例, 验证了方法的有效性和可行性, 为直接利用 C/C++ 开发 Android APP 提供了一种新途径.

关键词: Android 应用程序; Qt 开发框架; C/C++ 编程语言; Java 本地接口; 开发方法实践

引用格式: 闫锋欣, 牛子杰, 杜烁炜, 潘天丽. 基于 Qt 的 Android 应用程序 C/C++ 开发方法与实践. 计算机系统应用, 2018, 27(7): 96-102. <http://www.c-s-a.org.cn/1003-3254/6423.html>

Qt-Based C/C++ Development Pattern and Its Application for Android APP

YAN Feng-Xin, NIU Zi-Jie, DU Shuo-Wei, PAN Tian-Li

(College of Mechanical and Electronic Engineering, Northwest A & F University, Yangling 712100, China)

Abstract: Android Software Development Kit (SDK), a collection of Android tools, is an effective Java-based kit to develop Android Application (APP). However, Android APP's performance is usually limited by the backend Dalvik virtual machine environment. A novel practical model that Android APP developed by C/C++ codes is proposed, because Java Native Interface (JNI) can support C/C++ dynamic library callback mechanism. This paper firstly analyzes and compares the characters between the general development mode for Android APP and Qt cross-platform development framework, and presents a novel principle and a technical relationship between Android APP and Qt framework. Then, it describes the main processes, abstract implementation steps, and software list for the solution. Finally, the development of a sample APP, Mechanical CAD Teaching Assistant, is given as an example to verify the effectiveness and feasibility of the development pattern. This work provides a new way to use C/C++ programming language to develop APPs running on Android operating system.

Key words: Android application; Qt development framework; C/C++ programming language; Java Native Interface (JNI); development pattern and its application

随着移动互联网技术的飞速发展, 与智能手机应用程序 APP 开发相关的各类技术得到了不断发展和

深入应用, 开发了大量可在智能手机等移动终端运行的第三方应用程序 APP, 但存在总体数量丰富而现有

① 基金项目: 中央高校基本科研业务费专项资金(Z109021423); 农业部现代农业装备重点实验室开放课题(201603002); 国家自然科学基金(51302221); 西北农林科技大学教学改革项目 (JY1504046); 国家重点研发计划专项 (2017YFD0502000)

Foundation item: Special Foundation for the Fundamental Research Services for the Central Universities (Z109021423); Open Project of Key Laboratory of Modern Agricultural Equipment of Ministry of Agriculture (201603002); National Natural Science Foundation of China (51302221); Teaching Reform Project of Northwest A&F University (JY1504046); Special Program of National Key Research and Development Plan (2017YFD0502000)

收稿时间: 2017-10-31; 修改时间: 2017-11-27; 采用时间: 2017-12-07; csa 在线出版时间: 2018-06-27

基础功能代码重用率低下之间的矛盾. 因此, 为了能够有效地满足面向移动智能计算复杂 APP 对低性能硬件资源、实时信息交互服务的需求, 将广域分布的 Java、C/C++、Python 等源码和脚本进行按需组织和管理, 以实现移动智能计算平台 (Windows Phone、iOS、Android 等) 和桌面操作系统 (Windows、Mac OS、Linux/X11 等) 之间动态、跨自治域的资源共享与协同, 提高代码资源的综合利用率, 降低 APP 开发的技术难度和成本, 已成为一个重要的科学问题^[1].

为解决这一问题, 需要围绕移动智能计算平台 APP 开发模式建立能够支持代码资源共享和集成的技术方法. 近年来, 国内外学术界和工业界的研发工作取得了大量积极成果: (1) 从 APP 开发模式看, 主要有 Native 原生型、Web 网络型和 Hybrid 混合型三种^[2]. 其中, 原生开发是通过使用移动智能计算平台操作系统的 API 来实现方便、快捷、高性能调用终端的软硬件资源, 但存在学习难度大、移植性差等不足^[3]; 网络开发是依靠实时通信移动网络来实现 APP 的全部功能, 具有开发简单、速度快等优点, 但存在断网无法工作、硬件资源调用不便等缺点; 混合开发是借助第三方跨平台开发框架来实现 APP 的各项功能. (2) 从 APP 开发使用的语言看, 原生开发在 Windows Phone、Android 和 iOS 平台上会分别使用 C#^[4]、Java^[5]和 Objective-C^[6], 网络开发主要借助 CSS3、HTML5、JavaScript 等脚本与网络服务器的通信^[7,8], 而混合开发则会同时使用原生 API 的专用编程语言和网络脚本^[9].

Android 在移动智能计算领域中占有绝对的优势, Java 是其官方开发语言工具. Java 采用进程虚拟机的形式为 Android APP 提供虚拟运行环境, 并通过 APP 屏蔽底层硬件平台异构性的方式实现“一次编译随处运行”的跨平台特性^[10]. Norris 等^[11]利用 Java 和 Python 开发了可对大体量核磁共振数据进行并行化快速处理的 NMRFx; Radivoje 等^[12]借助 Java、JavaScript 和 HTML5 等开发了具有高级功能和强大表现力的车载信息娱乐系统; Gao 等^[13]采用 Java + SQL Server 开发了 Android 在线考试系统, 可实现制卷、判分、数据处理等考试环节的全过程自动化处理; 刘伟等^[14]利用 Android 智能手机和移动互联网设计开发了一种便携式角度测量仪, 具有良好的可操作性及便携性. 然而, 基于 Java 的 APP 运行时需依赖于底层 Dalvik 虚拟机环境, 需要较好的计算性能和硬件资源^[15,16]. 移

动智能计算和物联网的迅猛发展使这些问题在各类极端口应用中得到了逐级放大, 已成为 Java 难以逾越的障碍.

基于第三方跨平台框架的 APP 混合开发模式是上述问题的有效解决方案之一^[17]. PhoneGap、Titanium、jQuery 等现有的第三方框架多基于 CSS、HTML5、JavaScript 等脚本, 虽然解决了 APP 的跨平台问题, 但性能方面的提升并不明显. 而 Qt 跨平台编程框架的所有类都采用 C/C++ 开发, 底层类库封装了各平台的本地化实现, 因而具有极高运行效率, 并可广泛运行于 Windows、Mac OS、Android、Raspberry Pi 等桌面和移动智能计算设备平台^[18-20]. 此外, Qt 还支持 HTML5、Python 等解释型脚本和多种编程语言^[21].

1 Android APP 开发技术分析和框架设计

1.1 APP 开发技术分析

Android 移动智能计算平台上 APP 开发模式主要有 Native 原生型、Web 网络型和 Hybrid 混合型三种:

(1) Native 型 APP 的开发主要是借助运行平台自己的开发框架和 API. 例如, 在 Android 平台上, APP 是通过 Activity 来调用平台的底层 API 来执行各项功能, 而用户执行操作产生的交互结果、APP 运行时需要的各类参数等都会以 XML 形式存储起来.

(2) Web 型 APP 的开发主要体现在服务器端的环境配置. 在 Android 平台上, 每个 APP 都可简单地认为是由 APP 的用户界面 (GUI) 外加一个 Java Web 构成的, 全部功能的实现都依赖于实时网络.

(3) Hybrid 型 APP 的开发则主要是借助诸如 Apache Cordova 等的第三方框架进行开发. 这种开发形式的优点在于运行平台与硬件资源调用 API 无关, 缺点则包括调用性能差等.

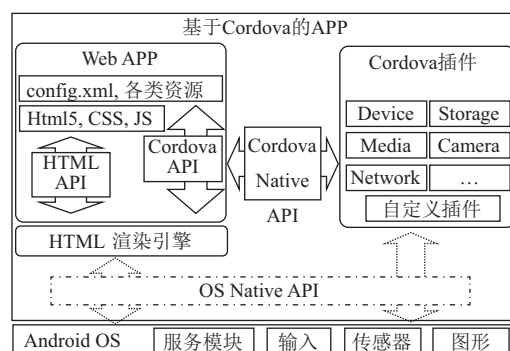


图1 Cordova APP 开发模式

这里以图1所示的Cordova第三方框架为例来进一步说明Android平台下APP开发的技术特点。Cordova开发APP有跨平台和面向平台两种开发方法,前者使用Cordova CLI(Command Line Interface)工具中的各类shell脚本集来管理APP所欲部署平台的配置内容,后者使用基于Web的Cordova组件集、Shell脚本和Plugman工具等对底层Android平台的SDK进行整合开发。可以看出,这种开发方法仍需借助CSS、JS和Shell脚本等才能够完成APP的开发。

1.2 基于C/C++的APP开发框架设计

针对重新学习Java等编程语言所带来的较高学习成本和技术风险问题,极有必要开展Android平台下基于C/C++的APP开发方法研究。这里给出基于C/C++的APP开发框架,如图2所示。



图2 基于C/C++的APP预期开发方法

图2中,采用C/C++开发APP的底层技术是解决C/C++和Java两类语言之间的相互调用问题,特别是针对Android平台的Java和C/C++类库之间相互调用的问题。从APP功能开发和实现的角度看,它的各种运算、控制等各功能模块的开发实现都用的是C/C++代码,就为重用其他项目的既有代码提供了直接入口,也为C/C++开发人员专注于新功能代码的编写提供了接口。同时,与用户交互的图形用户接口(GUI)、对Android底层API的调用、移动计算设备的硬件资源通信等衔接工作,则可以通过第三方的C/C++类库全权负责。

上述类库应具有的基本特点是:(1)全部采用C/C++编程语言;(2)具有与平台无关的跨平台能力;(3)能够满足移动智能计算平台硬件资源较低的苛刻要求。为此,本文选用已在桌面和嵌入式平台领域内广泛应用的跨平台开发框架Qt。下面探讨Qt开发Android APP的技术机制和实施方案。

2 Qt开发Android APP的工作机制

Qt是C/C++类库开发框架,因而必须先构建起与Java之间的相互调用通道。从Android APP的底层来

看,只能利用Google的NDK(Native Development Kit)来为基于C/C++的Android APP提供必要的编译工具和基础文件。同时,还要利用NDK来将C/C++程序封装嵌入到Android包文件(.apk)中,以及对Android SDK升级后的头文件和库文件进行调用。然而,由于Android APP独自运行于各自的Dalvik虚拟机中,且NDK只对外开放了虚拟机的部分接口,造成C/C++程序调试难度大、兼容性差等问题,需先开展其封装性研究。

2.1 基于Qt的Android APP封装模型的建立

利用Qt C/C++类库直接开发Android APP,需要分别考虑APP开发的两个部分:

- (1) 基于Qt类库的C/C++程序部分。这一部分主要是指APP实现数学计算、图像处理等的那些功能;
- (2) 基于Java的Android底层资源调用API部分。这一部分主要是指与Android底层硬件通讯和调用等的功能。

对于(1)来说,该部分的代码均是Qt C/C++的程序,在Qt内部会先经Qt元对象编译器(meta object compiler, moc)将含有emit、signals、slots、Q_OBJECT等关键字的Qt源代码处理为面向对象的标准C++类,然后再在不同平台下分别利用g++、gcc、msvc等编译器进行源代码的(交叉)编译,最后通过对Qt等各依赖库文件的链接,生成可在不同平台下运行的可执行APP程序。

对于(2)而言,该部分主要是指通过Java、XML等描述的和Android平台交互的各类资源,包括manifest、assets/strings/images、Ministro service和Java类等,用来实现APP在Android平台上的调用、权限获取、通信部署等工作。

为在(1)和(2)之间,即基于Qt的“纯”C/C++类源代码和不对第三方开放的Android平台之间,需以JNI(Java Native Interface)为桥梁来实现两者的协同工作。当Qt调用Android系统中的部分已有功能时,就可借助JNI访问它们。同时,Qt作为专注于跨平台的C++开发框架,也不可能全部实现Android的所有功能。某种程度上,JNI成为Android APP和C++程序之间相互调用的唯一通道。

为此,本文给出如图3所示的基于Qt的Android APP封装模型。其中,在“模块①”中,Qt会利用抽象层QPA(Qt Platform Abstraction)对全部操作系统平台进

行统一化支持, 确保采用 Qt 类库的所有程序均可直接运行于所有操作系统平台上, 且不限于 Android 平台; 在“模块③”中, APP 仍旧采用类似 Java APP 的形式, 依靠独自の Dalvik 虚拟机无缝运行于 Android 平台, 这样就可以通过所处的 Dalvik 虚拟机对第三方程序开放部分接口并支持外部的 JNI 调用; 当两者需要交互通信时, 可分别利用 Qt 事件循环和 Android 事件循环, 借助“模块②”中的类 THREAD 线程交换信息。

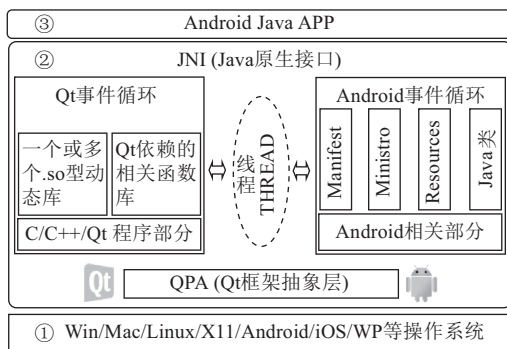


图3 基于 Qt 的 Android APP 封装模型

此外, 通过 Qt 对 Java 代码的支持, 可进一步拓展 Qt APP 的功能和应用范围, 把 Qt C/C++ 程序封装成 apk 包文件并平稳运行于 Android 平台上。

2.2 Qt C/C++与 Android Java 间的调用机制

要基于 Qt C/C++ 开发可运行于 Android 平台的 APP, 就必须先实现从 Qt C/C++ 程序调用 Java 的 Android 线程。这是因为, Android 的部分 API 设计机理不允许对外完全开放且必须在 Android 的 UI 线程中进行调用^[22]。然而, 因 Qt 不可能实现 Android 的全部功能而又要在 Qt 中调用这些功能时, 就需采用一定的方式来将其运行于 Android UI 线程中。

定义 1. 正向调用. 从 Qt C/C++ 事件循环线程到 Android Java UI 线程的调用称为正向调用。

定义 2. 逆向调用. 从 Android Java UI 线程到 Qt C/C++ 事件循环线程的调用称为逆向调用。

这里给出在 Qt C/C++ 与 Android Java 两者之间进行双向调用的多线程机制, 如图 4 所示。

该机制的正向调用主要过程如下:

步骤 1. 在 Qt C/C++ 事件循环线程中调用自定义的 Java 方法并在 Qt 线程中执行, 设该方法的名称为 `cppCallJava()`, 为在 Android Java 线程中访问 Android 的 API 做好准备。

步骤 2. 在 `cppCallJava()` 方法中, 借助 Android 的 `Activity.runOnUiThread`, 向 Android 的 UI 线程事件队列中添加一个 `public` 型 `Runnable` 动作并在 Android 的 UI 线程中执行这个 `Runnable` 动作。

步骤 3. 因处于执行状态的 `Runnable` 对象本已在 Android 的 UI 线程中, 因而就可以直接访问所需的 Android API。

若 Android 平台中的原生 Java APP 需调用基于 Qt 的 C/C++ APP, 就形成了逆向调用过程。此时, 与正向调用类似, 同样可在 Android UI 线程中调用 Qt 的 C/C++ 函数。

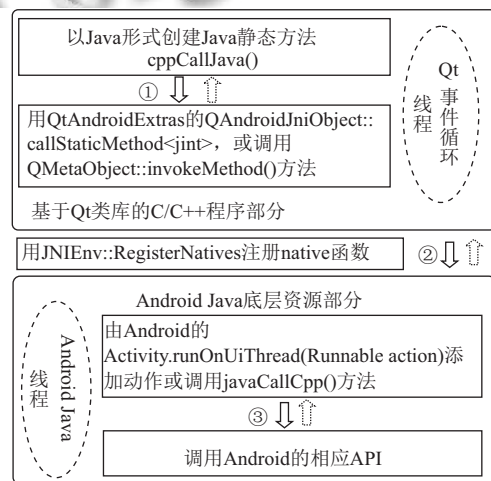


图4 Qt 与 Android 之间的技术关联

逆向调用的主要过程包括:

步骤 1. 在 Android UI 线程中调用相应的 Qt C/C++ 方法, 令其名称为 `javaCallCpp()`。

步骤 2. 在 `javaCallCpp()` 方法中, 使用 Qt 的 `QMetaObject::invokeMethod` 方法向 Qt 事件队列中添加一个异步或同步调用。

步骤 3. 在 Qt 线程中, 处理 `javaCallCpp()` 方法所添加的调用事件。

经过上述双向调用后, 就可以实现 Qt C/C++ 功能程序和 Android Java API 之间的相互调用了。

3 实验结果及展示

3.1 基于 Qt 的学生导教 APP 设计

图 5 给出的是机械 CAD 技术基础课程上机实践的基本流程。学生在实验室机房按照授课内容完成对应的上机编程练习, 用 U 盘等保存自己的代码。因各章

内容相对独立, 上机实践完成的习题源代码也相对分散, 仅有 30% 左右的学生能够在最后一次的综合实践课中重用这些代码; 同时, 由于机房内台式机完成的程序无法移植到学生的智能手机、平板等可移动智能计算设备上, 导致每次上机实验中主动编程的积极性不高, 相互复制抄袭的比例较高, 上机训练效果一般。此外, 因课程内容与实际的学习生活关联性不强, 学生缺少成就感。

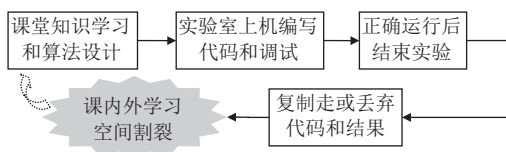


图5 传统机械CAD上机训练流程

针对上述问题, 将学生每次上机实践训练时完成的 C/C++ 程序代码作为 APP 的各项功能, 通过使用 Qt 构建的 APP UI 界面和 QPA 类屏蔽与 Android 平台相关的硬件资源具体差异, 再利用 Qt 的多线程机制在 C/C++ 事件和 Android Java 事件之间建立通信联系, 就可以开发出 Android 版的机械 CAD 导教 APP。整个 APP 的体系架构和各模块间的逻辑关系如图 6 所示。

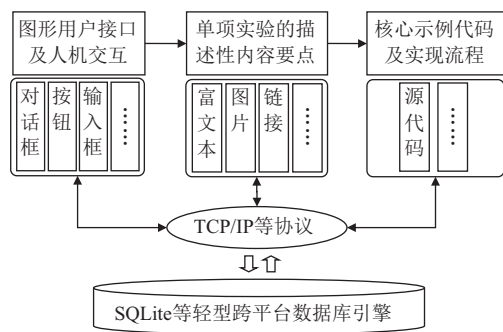


图6 Android版机械CAD导教APP的体系架构

APP 可实现的基本功能包括: (1) 在 APP 中查看不同章节的上机实验安排, 了解实验原理、算法步骤、注意事项等; (2) 点选单项实验后可通过完善模板空白或借助对话框、单(复)选框等进行人机交互; (3) 根据教师发布的新消息或内容更新, 由系统自动提示给相应的学生群组; (4) 点击“运行”等按钮后执行内置的单项实验程序样例, 为上机实验提供参考和指导; (5) 阅读、编辑、保存或另存修改后的核心程序代码。

3.2 导教 APP 的开发过程

本文以 Win7 为开发平台, 利用 Qt C/C++ 来完成

APP 的全部开发工作。基于 Qt C/C++ 开发框架中的 Qt Designer、uic、moc 等辅助开发工具, 可以把开发的用户界面翻译为 XML 中间文件, 再通过内部编译连接工具翻译为可在 Android 平台下运行的二进制文件。同时, 为了能够方便地“一键”实现从 C/C++ 源代码开发环境直接到 Android 平台上 APP 的自动运行, 还可以使用 Ant、Gradle 等自助打包开源软件。为此, 将完成各部分辅助开发功能的(开源)软件整理后列示在表 1 中。

表 1 Win7 平台 APP 开发所需的辅助工具

工具名	文件名全称
Qt	qt-opensource-windows-x86-5.8.0.exe
JDK	jdk-8u121-windows-i586.exe
Ant	apache-ant-1.10.0-bin.zip
Gradle	gradle-3.0-all.zip
NDK	android-ndk-r11b-windows-x86.zip
SDK	android-sdk-installer_r24.4.1-windows.exe
ADB	adb.zip

具体来说, 各软件的主要作用是: (1) Android SDK, 即 Software Development Kit(Android 软件开发工具包), 采用 Java 开发完成, 是 Android APP 运行所需的底层基础组件和 Android 平台开发的专属工具包; (2) JDK, 即 Java Development Kit(Java 开发工具包), 包括 Java 运行环境 (Java 虚拟机与 Java 系统类库) 和 Java 基本组件工具; (3) Apache Ant 或者 Gradle 可生成 APP 的 .apk 包文件, 可自动完成编译、测试和部署等步骤; (4) ADB, 即 Android Debug Bridge(Android 调试桥), 可在电脑上调试 Android 手机设备或模拟器; (5) Google NDK, 即 Native Develop Kit(原生开发工具包), 支持 Android 平台调用 C/C++ 程序。

基于 Qt C/C++ 的 Android APP 开发步骤主要包括:

步骤 1. 安装所需开发工具软件并对其进行配置。各辅助开发工具软件的安装配置要遵循先安装再配置的方式来有效解决相互之间的依赖关系, 如表 1 中的各软件工具可在依次安装 Qt、JDK、Apache Ant、Gradle、Android NDK、SDK 和 Google ADB 后, 对它们的运行环境变量进行相应的设置。

步骤 2. 利用 QtCreator 中内嵌的 Qt Designer 开发 APP 的用户界面, 在相应的功能模块区域中填充相应的按钮、输入框等窗口部件, 设计过程中随时可以使用 Alt+Shift+R 快捷键等方式预览设计效果, 如图 7 所示。

步骤 3. 通过添加不同的信号-槽连接, 设置所需的 JNI 静态函数, 将界面元素和相应的功能实现源代码关联起来, 达到点击按钮产生相应动作响应的效果; 同时, 为加速开发过程和减少功能优化时的复杂性, 可先采用 Desktop 桌面形式进行开发调试, 并综合利用 QtCreator 集成开发环境中的编译、连接功能对 APP 系统进行优化、测试和调试。

步骤 4. 达到 APP 设计的满意效果后, 将 QtCreator 中运行模式调整为 Android for armeabi-v7a, 然后进行编译、连接和 .apk 包在 Android 操作平台上的自动安装. 生成的 .apk 安装包文件既可以通过手动方式在 Android 手机上安装, 也可通过配置 QtCreator 开发环境后自动安装。

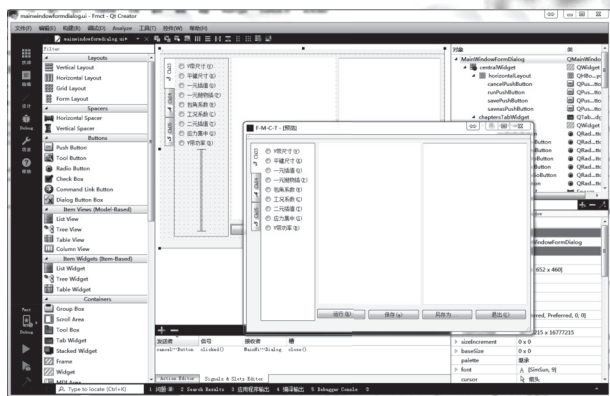


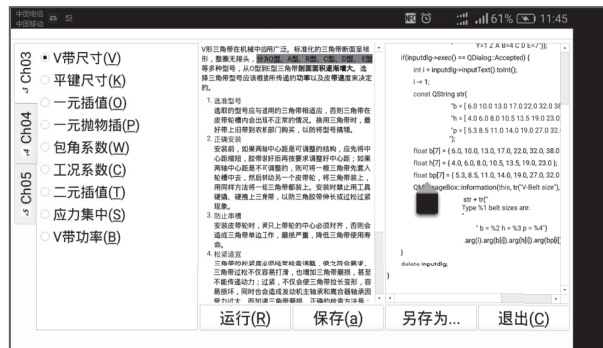
图 7 APP 软件系统在桌面环境下的开发效果

3.3 结果展示与评价

图 8(a) 是 Android 版机械 CAD 课程导教 APP 在手机 (型号 HUAWEI MT7-CL00) 横放时的显示效果, 用户此时选定了计算“V 带尺寸”实验. 在 APP 界面的左侧, 给出了各章上机实践所需完成的任务, 还包括“平键尺寸”、“一元插值”、“V 带功率”等实验项目; 界面中间是每个实验的说明, 未显示完全的文字可通过边侧的滑动条进行翻页; 界面右侧是该项实验所需的核心代码。

图 8 中的黑色区域是手机屏幕的上方和右侧, 用来显示 Android 手机的系统菜单栏. 值得注意的是, 只有当如图 8(a) 所示的形式, 对 APP 中的叙述性文字或者代码进行了修改时, 按钮“保存”和“另存为…”才会变为可用状态. 图 8(b) 给出的是在点击了图 8(a) 中的按钮“运行”后, 查询出的 Z 型 V 带 (其代号为 1) 的截面尺寸值; 图 8(c) 是同一计算过程在 Win7 操作系统下

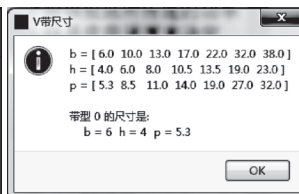
的显示效果. 可以看出, 同一对话框因运行平台所能够支持的默认参数不同而具有一定的显示效果差异。



(a) 运行中修改APP的显示字符



(b) Android查询结果



(c) Win7查询结果

图 8 Android 6.0 下 APP 运行效果及查询结果对比

在 2016–2017 学年第 1 学期 (2016 年 10 月 17 日~12 月 25 日) 的机械 CAD 课上机实践训练过程中, 由机械专业的 122 名本科生首次使用 Android 版机械 CAD 导教 APP 进行了性能测试和实验评估. 图 9 给出了全体学生使用频率的时间分布, 可以看出, 从第 8 周课程开始到第 17 周课程结束, 10 周期间共收到学生提交更新建议 19 项、疏漏补正 76 个, APP 打开使用 6025 次 (网络连接状态). 以 10 周共 50 个工作日为例, 每天人均使用 0.988 次, 说明在日常教学活动中得到了较好应用. 从第二周的周五开始, 每周五都有较高的使用次数, 这也与周五是上机实践训练课的实际情况相符. 在周一至周四的日常学习期间使用量并不多, 可见本科生并不愿意在专业课上投入过多的学习时间和精力。

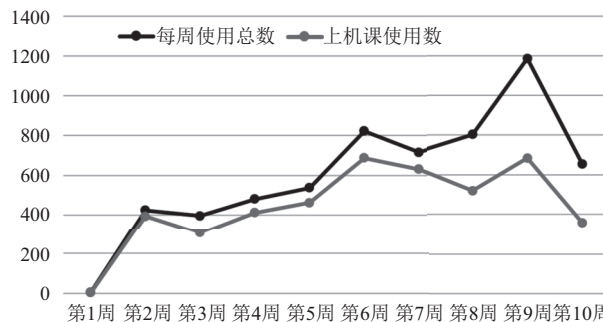


图 9 APP 的每周使用数分布

可以看出,利用 C/C++编程开发语言并借助 Qt 开发框架,通过使用相关开源软件能够开发出满足实际需要的 Android APP 应用程序,既降低了重新学习 Java 和 JavaScript 等开发环境所带来的负担,又为重用户已有的 C/C++程序提供了新途径。

4 结论与展望

本文在分析了 Google 的 NDK 不能为 C/C++程序提供全部 Android 功能调用的基础上,给出了基于 Qt C/C++框架开发 Android APP 的方法和原理,深入阐释了 Qt C/C++与 Android Java 两者之间的调用方法,给出了 Win7 平台下开发 Android APP 过程的主要步骤。同时,通过 Android 版机械 CAD 导购 APP 的开发和该 APP 的实际使用结果的展示和评价,说明了基于 Qt C/C++开发 Android APP 方法的有效性和可行性。在未来的研究工作中,将进一步对复杂需求环境(如图形快速渲染等)下基于 Qt 和 Java 开发的 Android APP 综合运算性能做出分析和评测。

参考文献

- 1 Latif M, Lakhri Y, El Habib Nfaoui, *et al.* Cross platform approach for mobile application development: A survey. Proceedings of 2016 International Conference on Information Technology for Organizations Development (IT4OD). Fez, Morocco. 2016. 1-5.
- 2 朱春华, 张宜志. 粮食储备与决策系统的移动终端 APP 开发模型研究. 食品工业, 2017, 38(4): 254-256.
- 3 李亚东, 冯仲科, 曹明兰, 等. Android 智能手机树高测量 APP 开发与试验. 中南林业科技大学学报, 2016, 36(10): 78-82.
- 4 滕健, 万福成. 基于增强现实的产品展示 APP 设计研究. 包装工程, 2017, 38(14): 219-223.
- 5 杨林楠, 郜鲁涛, 林尔升, 等. 基于 Android 系统手机的甜玉米病虫害智能诊断系统. 农业工程学报, 2012, 28(18): 163-168. [doi: 10.3969/j.issn.1002-6819.2012.18.024]
- 6 冷伟锋, 马占鸿. 小麦条锈病移动端监测平台的构建. 中国植保导刊, 2015, 35(5): 46-49, 84.
- 7 黄悦深. 基于 HTML5 的移动 Web APP 开发. 图书馆杂志, 2014, 33(7): 72-77.
- 8 刘白林, 王鹏博. PhoneGap+HTML5+SPA 微信公众平台实现. 西安工业大学学报, 2017, 37(6): 448-452.
- 9 Dalmaso I, Datta SK, Bonnet Christian, *et al.* Survey, comparison and evaluation of cross platform mobile application development tools. Proceedings of the 9th International Wireless Communications and Mobile Computing Conference. Sardinia, Italy. 2013. 323-328.
- 10 怀进鹏, 李沁, 胡春明. 基于虚拟机的虚拟计算环境研究与设计. 软件学报, 2007, 18(8): 2016-2026.
- 11 Norris M, Fetler B, Marchant J, Johnson BA. NMRfx processor: A cross-platform NMR data processing program. Journal of Biomolecular NMR, 2016, 65(3-4): 205-216. [doi: 10.1007/s10858-016-0049-6]
- 12 Ostojic R, Pesic J, Bjelica MZ, *et al.* Java-based graphical user interface framework for in-vehicle infotainment units with WebGL support. Proceedings of the 6th International Conference on Consumer Electronic-Berlin (ICCE-Berlin). Berlin, Germany. 2016. 184-186.
- 13 Gao SF, Xie YH. The design and implementation of a cross-platform teaching system. Proceedings of the 3rd International Conference on Learning and Collaboration Technologies. Toronto, ON, Canada. 2016. 303-309.
- 14 刘伟, 李杰, 杨雁宇. 基于 Android 的便携式角度测量仪设计. 电子器件, 2017, 40(4): 1044-1048.
- 15 Corral L, Sillitti A, Succi G. Mobile multiplatform development: An experiment for performance analysis. Procedia Computer Science, 2012, (10): 736-743. [doi: 10.1016/j.procs.2012.06.094]
- 16 Wagner M, Blumenstein K, Rind A, *et al.* Native cross-platform visualization: A proof of concept based on the Unity3D game engine. Proceedings of the 20th International Conference Information Visualization (IV). Lisbon, Portugal. 2016. 39-44.
- 17 Latif M, Lakhri Y, El Habib Nfaoui, *et al.* Review of mobile cross platform and research orientations. Proceedings of 2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems. Fez, Morocco. 2017. 1-4.
- 18 Blanchette J, Summerfield M. C++ GUI Qt4 编程. 2 版. 闫锋欣, 译. 北京: 电子工业出版社, 2008.
- 19 Cale D, Chimbo V, Paz-Arias H, *et al.* People recognition for loja ECU911 applying artificial vision techniques. Latin American Journal of Computing Facultad de Ingeniería de Sistemas, 2016, 3(1): 27-34.
- 20 喻晓, 夏澎. 基于 Qt 的便携式心电图监护仪 GUI 应用软件设计. 计算机系统应用, 2017, 26(5): 221-226. [doi: 10.15888/j.cnki.csa.005766]
- 21 Summerfield M. Python Qt GUI 快速编程: PyQt 编程指南. 闫锋欣, 黄琳雅, 王军锋, 等译. 北京: 电子工业出版社, 2016.
- 22 陈美镇, 王纪章, 李萍萍, 等. 基于 Android 系统的温室异构网络环境监测智能网关开发. 农业工程学报, 2015, 31(5): 218-225.