

文章编号: 2096-1472(2017)-10-17-03

一种跨平台移动APP开发方法的研究及实现

曹蔚然, 杜梅, 沈忠起, 杨彬彬

(沈阳师范大学软件学院, 辽宁 沈阳 110034)

摘要: 本文针对移动平台APP开发需同时兼顾多种平台的现状, 对一种跨平台移动APP开发方法——React Native方法进行了初步研究。文章介绍了该方法提出的背景、解决问题的思路、用到的编程语言和程序库的软件架构, 指出了该方法的巧妙之处、优越性, 以及所用语言的发展趋势, 最后给出了用该方法实现的跨平台移动APP系统的一个模块实例, 并结合模块程序代码和界面讲解了该方法实现的一些细节。系统的实现证明了这种跨平台移动APP开发方法的有效性和实用性。

关键词: 跨平台; 移动APP开发; React Native

中图分类号: TP399 **文献标识码:** A

Research and Implementation of a Cross-Platform Mobile Application Development Method

CAO Weiran, DU Mei, SHEN Zhongqi, YANG Binbin

(Software College, Shenyang Normal University, Shenyang 110034, China)

Abstract: Based on the current situation of mobile application development, this paper conducts a preliminary study on a cross-platform mobile application development method, Reaction Native. This paper introduces the background, the solving thoughts, the programming language and the library software architecture of this method. This paper points out the advantages of the method and the development trend of the language used. At last, the paper presents a module instance of the cross-platform mobile application system implemented with this method, and some programming details are introduced by combining the module program code with the interface. The implementation of the system proves the effectiveness and practicability of this cross-platform mobile application development method.

Keywords: cross-platform; mobile application development; React Native

1 引言(Introduction)

当今移动平台APP开发炙手可热, 而且温度还在不断攀升。在安卓成为世界第一大移动平台后, 主流的移动平台俨然成为安卓和苹果的天下。安卓平台用户人数众多, 市场巨大, 但其免费政策使得盈利成为一个值得思考的问题; 苹果平台使用者在世界范围内人数虽少, 但在美英等发达国家却反超安卓, 同时其收费政策使得其使用者大多“不差钱儿”, 所以盈利又似乎是唾手可得的事情。选安卓还是选苹

果是摆在每一个移动平台开发团队面前的两难选择, 更别还有其他的移动平台, 如微软的Windows Phone。两大主流移动平台API大相径庭, 学习曲线都较陡峭, 同时各平台API更新换代频繁, 新技术、新问题层出不穷, 所以想同时开发两个平台的软件就得维护两支开发队伍, 这种“事倍功半”的“奢侈”开发方式并不是一般规模的开发团队能承受得了的事情。于是寻找一种能有效节省人力、物力的跨平台APP开发方法成为移动平台APP开发中一个值得探索的问题。

2 跨平台开发(Cross-Platform development)

跨平台的概念由来已久。当年，为了解决不同操作系统间软件移植的问题，Java提出了JVM的概念。一个虚拟机很好地解决了同一段代码在不同平台上运行出相同效果的问题。然而虚拟机是有代价的，即效率。虚拟机需要解释中间代码来运行软件。相比于机器码直接运行，这种解释运行效率要低很多，显现出来就是软件运行反应慢^[1]。这种效率的降低同样体现在以HTML为核心的Web上。Web也是跨平台的，但执行效率与本地程序相比不可同日而语，而且Web访问本地资源有着安全上的先天制约^[2]。

移动平台计算资源紧张，这些问题显得更为突出，所以直接应用Java和Web到移动平台是不行的。于是一个变通的被称为混合(Hybrid)的方法被提出。该方法是通过本地的一个软件框架来运行远程Web内容并加以交互以达到访问本地资源的目的。但这种方法与本地程序相比，首先是执行效率低的问题，其次是用户界面重复开发且与本地程序不易一致的问题^[3]。另一种方法是通过使用一种通用的编程语言进行编程，然后生成不同移动平台的本地程序的办法来解决跨平台的问题^[4]。这种方法有很大的优势。首先，虽然会增加少量辅助程序，但因为生成的是本地程序，其执行效率总体来说与本地原生程序一样；其次，由于其使用的是本地的UI，所以界面与本地原生程序完全一致。这种方法中React Native是最早提出也是目前最好的解决方案^[5,6]。

3 React Native方法(The react native method)

React Native是应用React Native来进行移动跨平台开发的方法。React Native是由Facebook公司在2015年推出的一组程序库，用以把其JavaScript库React引入主流移动平台的开发中。这之前其React库已经在Web UI领域取得相当大的成功。React Native的推出改变了人们对移动平台开发的固有观念，为移动跨平台开发注入了新的活力^[5]。

3.1 JavaScript

React Native主要使用JavaScript语言进行编程。JavaScript语言是一种在Web初期就已经存在的脚本语言。它简单易学，功能强大，早期便吸引了大量使用者。但由于最初其仅是作为一种Web动态化的辅助语言出现，因此使用者虽然多，但大多是业余程序员或其他行业的爱好者，并没有引起多少专业程序员的重视。然而，JavaScript神奇的地方在于其发展，总是能平凡中出神奇，一直处在计算机语言发展的潮头浪尖。2005年，Ajax概念被提出。这种神奇的编程方式在当年立即就刷爆了大众的眼球，让人们知道原来网页编程还可以这样做。这之后其发展可谓一发而不可收，不但

不断涌现出各种神奇的框架和程序库，其语言本身也发展迅速，许多最新的编程理念都能在JavaScript中得以体现。时至今日，JavaScript已发展到计算机的许多领域，不但能做其他编程语言能做的事，而且还通常做得更好，其跟随者中专业程序员大量增加，编出的各种应用、框架及库的质量越来越高，俨然已成为最受欢迎的编程语言之一。JavaScript能取得今天的的成绩和许多因素有关，但让其走出网页走向更多领域却与一个项目有着直接的关系，即V8项目。V8项目是一个JavaScript引擎项目，它能让JavaScript以接近甚至等于本地程序的效率得以执行，这直接扩大了JavaScript的应用范围并产生了绝对的竞争力。以此为基础加之Node.js的出现直接促进了React Native的产生^[7]。

3.2 React Native架构

React Native抓住当前移动平台开发的特点，很巧妙地利用各移动平台自身的优势来完成跨平台开发任务，避免了“重复制造轮子”。其程序架构情况如图1所示。APP程序一般通过React来访问React Native，然后通过JavaScript运行时库来访问底层移动操作系统以达到调用本地组件运行的目的^[8]。当然程序的运行还有多种渠道可灵活选择，以满足丰富的程序要求。这一过程中APP首先要运行本地系统支持的语言(在安卓中是Java，在苹果中则是Object-C)，然后本地语言通过中间桥与JavaScript容器通信，最终完成与JavaScript的交互，如图2所示。



图1 React Native架构图

Fig.1 React Native architecture

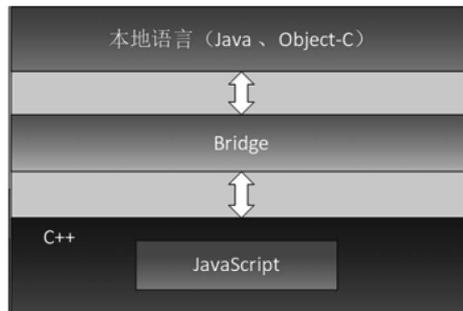


图2 React Native程序通信机制图

Fig.2 React Native software communication

4 系统实现(System implementation)

React Native的编程语言采用的是JavaScript的扩展版本JSX,且目前官方支持JavaScript的标准ES6。其程序以“.js”为扩展名,各模块间用输入(import)、输出(exports)语句进行代码复用,内部建立页面的虚拟DOM,用State进行维护,并用render()函数生成界面。下面以登录模块为例来看一下系统在安卓上的实现效果。

登录模块涉及了React Native界面装入、控件显示及界面切换的方法。系统启动时,从根目录的App.js开始执行,在其中需要安排好系统启动前的准备工作(如大图片的异步装入)并指定具体切入的界面位置。异步装入代码如下:

```
01:  _loadResourcesAsync=async()=>{
02:    return Promise.all([
03:      Asset.loadAsync([ //异步装入图片
04:        require('./assets/images/mail1.png'),
05:        require('./assets/images/lock1.png')
06:      ]),
07:      Font.loadAsync([ //异步装入字体
08:        Ionicons.font,
09:        {'space-mono':require('./assets/fonts/SpaceMono-Regular.ttf') }
10:      ]),]);};
```

代码中使用了React Native库expo的Asset模块来进行异步装入以提高后续程序访问资源的效率。

进入切入页面后需要建立导航器(Navigator)以进行后续页面切换。代码如下:

```
01: export const RootStackNavigator=StackNavigator(
02:   {Lg:{screen:LgScreen,}, //登录页面
03:   PageFace:{screen:PageFace,}, //内容页
04:   Settings:{screen:SettingsScreen,}, //设置页
05: }
06: );
```

代码中建立的导航器共有三项,分别是Lg、PageFace、Settings,分别指向三个页面。在程序中需要切换页面时只使用如下代码即可:

```
this.props.navigation.navigate('目标页面')
```

登录页面及其对应代码的一些情况如图3所示。用户名和密码输入框使用<TextInput>组件,密码框需设定其secureTextEntry属性为true,前方的图标需用设置该行容器样式为行布局的方式来进行固定。

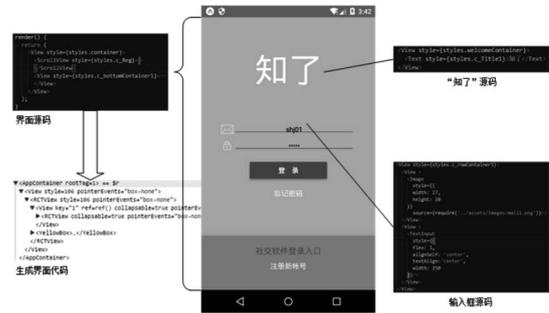


图3 登录界面及对应代码

Fig.3 Login screenshot and code

5 结论(Conclusion)

使用React Native进行移动平台跨平台APP开发,可以用简单、高效、用户广的JavaScript语言编写程序,一次编写可生成多种主流移动平台程序。生成的程序直接调用本地组件,其效率在理论上与本地程序基本一样。这样不但大大节省了系统开发的时间和人力、物力,进而减小开发成本,而且开发出来的程序执行效率高,是一种有效和高效的移动平台APP开发方法。

参考文献(References)

- [1] Manson J,Pugh W,Adve S V.The java memory model[J].Acm Sigplan Notices,2005,40(1):378-391.
- [2] Gensing-Pophal L.HTML5[J].Econtent,2011,34(2):10.
- [3] Ghatol R,Patel Y.Beginning PhoneGap:Mobile web framework for JavaScript and HTML5[M].Apress,2012:1-16.
- [4] Eisenman B.Writing cross-platform apps with react native[EB/OL].<https://www.infoq.com/articles/react-native-introduction>,2016-02-25.
- [5] Wikipedia.React(JavaScript library)[EB/OL].[https://en.wikipedia.org/wiki/React_\(JavaScript_library\)](https://en.wikipedia.org/wiki/React_(JavaScript_library)),2017-09-28.
- [6] Occhino T.React Native:Bringing modern web techniques to mobile[EB/OL].<https://code.facebook.com/posts/1014532261909640/react-native-bringing-modern-web-techniques-to-mobile>,2015-02-08.
- [7] Wikipedia.JavaScript[EB/OL].<https://en.wikipedia.org/wiki/JavaScript>,2017-09-26.
- [8] Lakshmanan S.Introduction to React Native[EB/OL].<https://www.slideshare.net/sambhu7/introduction-to-react-native-71847255>,2017-02-7.

作者简介:

曹蔚然(1974-),男,博士,讲师.研究领域:网络安全,软件开发,智能算法.

杜梅(1977-),女,博士,讲师.研究领域:软件开发,数字图像处理.

沈忠起(1997-),男,本科生.研究领域:软件开发.

杨彬彬(1993-),男,本科生.研究领域:软件开发.