

基于 Android 的通讯录实时同步功能^①

陆少鹏, 周渊平

(四川大学 电子信息学院, 成都 610065)

摘要: 随着 Android 系统的不断发展, Android 系统被应用在各种设备上面, 包括将 Android 系统应用到有线电话上. 因此需要开发一个通讯录能够在搭载了 Android 系统的有线电话和手机之间实现数据实时同步. 通讯录的客户端和服务端是通过 Socket 来建立连接的, 然后采用 Handler 机制发送数据和读取数据, 实现了通讯录数据的实时同步. 经过测试, 通讯录实现了手机与有线电话的通话记录, 增加、修改和删除联系人的实时同步功能.

关键词: Android; 通讯录; 实时同步; socket; Handler

Real-Time Synchronization Function for Contact Based on Android

LU Shao-Peng, ZHOU Yuan-Ping

(College of Electronic Information, Sichuan University, Chengdu 610065, China)

Abstract: With the continuous development of Android system, the Android system has been used in a variety of devices, including applying to wire telephones. Therefore, we need to develop a contact which has the real-time synchronization function between wire telephone and mobile phone equipped with Android system. The contact establishes connection between client and server by using Socket, and using Handler mechanism to send data and read data, as a result, it realizes the function of the real-time synchronization for contact data. The test shows the contact achieves the goal of the real-time synchronization function of calling records, adding, modifying and deleting contacts between mobile phone and wire telephone.

Key words: Android; contact; real-time synchronization; socket; Handler

日常办公通讯通常使用固定电话, 它具有抗干扰能力强, 通话质量好, 保密性高的特点, 最突出的是辐射小^[1], 因此在室内办公时, 用户更加趋向于使用有线电话. 在这种情况下, 将手机和有线电话的通讯录结合开发, 实现在室外办公时能够实时同步有线电话的未接来电信息, 方便即时回复未接来电, 在室内办公时也能够实时同步手机的未接来电和联系人信息, 方便电话拨打.

目前的通讯录同步是为了解决一个用户的不同终端设备的用户数据备份不一致给用户带来不便的问题, 需要用户手动上传和下载用户数据才能够实现通讯录同步^[2]. 现在 Android 系统的通讯录同步比较典型的应用是小米云同步服务, 其包括了通话记录同步、便签

同步和短信等用户数据的同步^[3], 但是小米云同步也需要手动上传和下载数据进行数据同步, 并不是自动进行同步的, 而且是实现手机与手机之间或者是手机与平板电脑之间的数据同步.

本文所做的通讯录的实时同步和现在的通讯录同步不同的是, 该通讯录实现的是实时的和自动的通讯录数据同步, 因此实时性比现有的通讯录同步好, 由于是自动实现通讯录数据同步, 所以该通讯录同步操作比现有的通讯录同步简便. 而且同步的是手机和有线电话的通讯录数据库数据方便室内外的办公. 本文所介绍的有线电话是由搭载了 Android4.0 操作系统的 TQ210 开发板实现的, 具备了拨号的功能.

① 收稿时间:2016-08-03;收到修改稿时间:2016-09-18 [doi: 10.15888/j.cnki.csa.005725]

1 Android系统架构

Android 系统架构采用了分层架构,从高层到低层依次分为应用程序层、应用程序框架层、系统运行库层和 Linux 核心层^[4,5],如图 1.

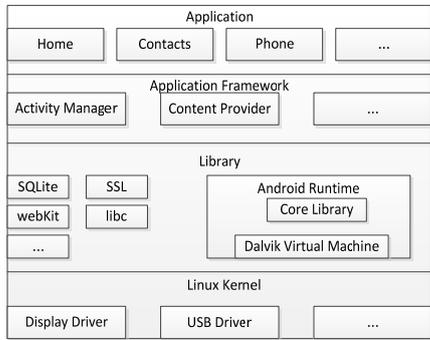


图 1 Android 系统架构图

应用程序层所有的程序都是 JAVA 语言所编写,通过调用应用程序框架层所提供的 API 来完成的^[6]. Android 的系统运行库层分为程序库和 Android 运行库.程序库主要包含一些 C/C++库,这些库能够被 Android 系统的不同组件使用,它们通过应用程序框架层给开发者提供服务. Android 运行库提供 Java 变成语言核心库大部分功能. Android 的 Linux 核心层是基于 Linux2.6 内核的,是硬件和软件之间的抽象层.

2 通讯录实时同步系统框架

通讯录同步系统主要分为客户端和服务端,它们之间通过 Socket 建立连接.常用的 Socket 有两种:流式 Socket 和数据报式 Socket.流式 Socket 是一种面向连接的 TCP 服务应用,数据报式 Socket 是一种无连接的 UDP 服务应用.通讯录的实时同步功能的客户端和服务端是采用了流式 Socket,其原理框图如图 2 所示.

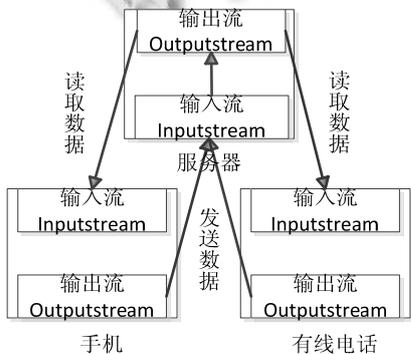


图 2 系统框图

由图 2 可知,通讯录的实时同步分为两个方向,分别为:手机端实时同步有线电话的通讯录数据和有线电话实时同步手机端的通讯录数据.

当有线电话端的通讯录数据库发生变化时,有线电话端就会将发生变化的数据转换成输出流,并将该输出流发送给服务器端,服务器端读取到该输出流数据后将它转换为输出流供手机端读取该同步数据,该过程实现了手机端实时同步有线电话端的通讯录数据,如图 2 中灰色线过程.

有线电话端实时同步手机端通讯录数据的过程就是红色线过程,即当手机端的通讯录数据库发生变化时,手机端就将变化的数据转换成输出流并将该输出流发送给服务器端,服务器端读取到该数据后再将该数据转换为输出流供有线电话端读取同步数据,最终实现有线电话通讯录实时同步手机端的通讯录数据.

3 客户端

通讯录的实时同步功能主要包括四个部分:通话记录、添加联系人、删除联系人和修改联系人的实时同步.客户端的程序流程图如图 3.

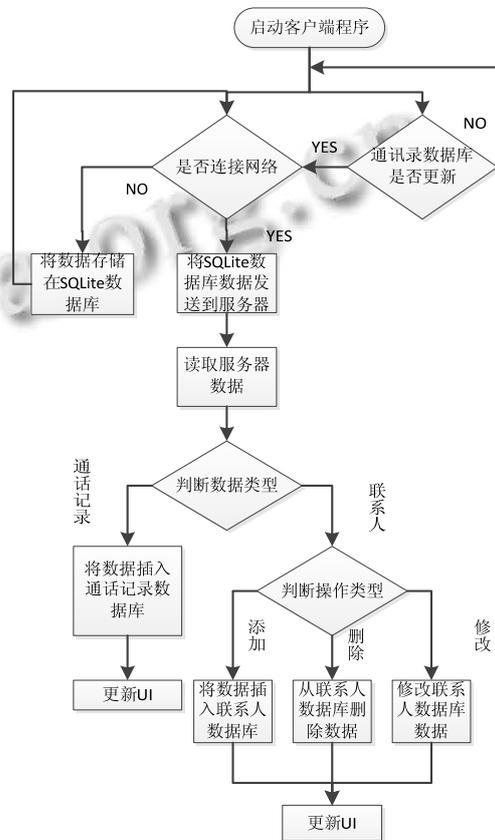


图 3 客户端程序流程图

客户端的实现主要分为两个部分,一个部分是判断客户端是否连接网络,连接网络就将同步数据发送到服务器端,未连接网络就将同步数据存到 SQLite 数据库中.另一部分就是要实时监听通讯录数据库的数据变化,有数据变化时就发送数据到服务端,没有数据变化时就处于不断监听状态.

客户端为了能够将同步数据发送到服务器端需要处于连接网络的状态,所以,在发送数据之前必须先检查客户端是否连接网络.客户端用 `getSystemService(Context.CONNECTIVITY_SERVICE)` 方法来获取 `ConnectivityManager` 类的对象,然后再通过该对象的 `getAllNetworkInfo()` 方法来获取到客户端的所有网络连接情况的对象,最后再使用该对象的 `getState()` 方法来判断客户端网络连接情况并且与 `NetworkInfo.state.CONNECTED` 对比判断是否相等就可以判断出客户端是否处于连接网络的状态.如果上面的判断结果是客户端未连接网络,则将同步数据存储到 SQLite 数据库中.SQLite 数据库是通过继承 `SQLiteOpenHelper` 类来实现的.该数据库里面创建了两个表,分别为联系人表和通话记录表,分别用于存储联系人同步数据和通话记录同步数据.如果客户端判断为处于连接网络状态,则先将 SQLite 数据库中存储的所有同步数据和客户端数据库刚更新的同步数据发送到服务器端.每当将 SQLite 数据库中的同步数据发送出去之后,就会调用 SQLite 数据库的 `delete()` 方法将旧的数据删除掉,方便之后的操作.

要实现数据的实时同步首先得通过使用 `ContentObserver` 内容观察者监听通话记录数据库数据和联系人数据库的变化.假设手机有来电或则去电致使手机端的通话记录数据库数据发生变化或者手机端有对联系人数据库进行增加、删除或修改的操作致使联系人数据库发生变化,就会触发 `ContentObserver` 里面的 `onChange()` 方法,然后就可以在这个方法里面去获取新的数据并且通过 `out.println()` 把这条数据发送给服务器端.同时,有线电话端就可以通过重写 `Handler` 机制的 `handleMessage()` 方法来从服务器端获取到同步数据.有线电话端获取到同步数据之后,读取同步数据的标志位 `flag` 来判断该条数据是属于通话记录类型的还是联系人数据类型的.如果该条数据属于通话记录数据类型的,就需要将同步数据与本地通话记录数据库数据进行对比,如果该条同步数据是本地数据则

不进行任何操作,如果同步数据非本地数据就需要通过 `Util.AddNumToCallLog()` 把同步数据插入到通话记录数据库中.如果这条数据是联系人数据类型的,也需要将该条同步数据与本地联系人数据库数据进行对比,如果该条数据属于本地数据库则不进行任何操作,否则通过 `Util.AddContact()`、`Util.ChangeContact()` 和 `Util.DeleteContact()` 分别对联系人数据库进行添加、修改和删除操作.通话记录数据库或则联系人数据库完成数据更新之后,就会触发加载器 `Loader` 的回调机制 `onLoadFinished()` 方法通知最终的运行结果,之后就会调用适配器的 `notifyDataSetChanged()` 方法,即当适配器的内容发生变化时通过这个方法强制调用 `getView` 来刷新每个 `item` 的内容,可以实现动态刷新列表从而动态更新 UI 界面.通过以上步骤有线电话端就可以实现实时同步手机端的数据了,反之同样可以实现手机端实时同步有线电话端的数据.

客户端的 Android 程序是在 Eclipse 中建立的,在工程中建立了通话记录功能包、联系人功能包和内容观察者功能包等,这些功能包里面的 Java 代码分别实现了通讯录的联系人功能和显示通话记录功能等,其 Android 工程目录如图 4 所示.

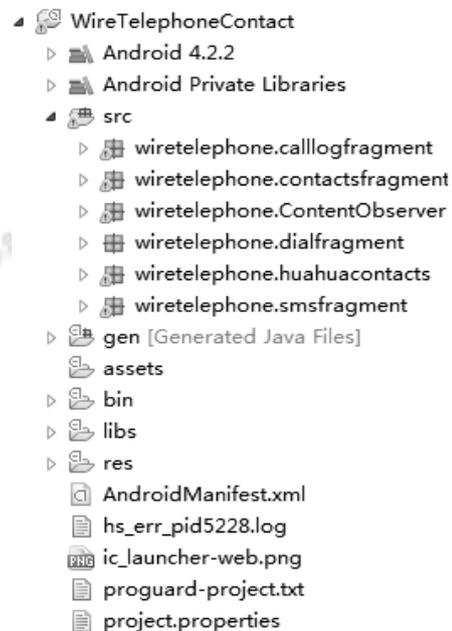


图 4 客户端工程目录

4 服务器端

服务器端是采用 `ServerSocket` 创建的 TCP 服务端.服务器端程序流程图如图 5 所示.

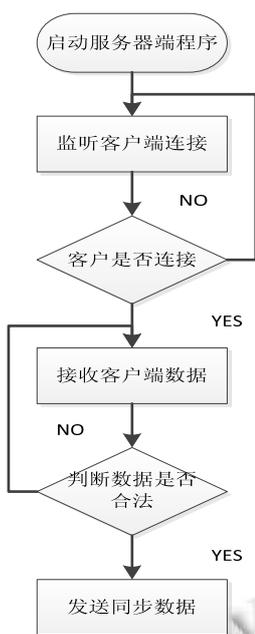


图 5 服务器端程序流程图

服务器端为了实现与客户端的连接, 首先得使用 `ServerSocket()` 构造器创建一个服务器端的对象, 然后在 `while(true)` 代码块里面使用 `server` 类的 `accept()` 方法不断地监听等待来自客户端的连接请求。由于, 服务端得实现来自多个客户端的连接请求, 所以, 得使用 `Executors` 类下面的 `newCachedThreadPool()` 方法来存储多个客户端的连接请求并且得使用 Java 的多线程技术来分别处理来自不同客户端的连接请求。当服务器端监听到了来自客户端的连接请求之后, 就会和客户端建立连接并且将该客户端添加到连接池中。与客户端建立连接之后, 为了能够获取到客户端的同步数据, 需要在服务器端使用 `Socket` 对象的 `getInputStream()` 方法来获取输入流对象, 并且使用输入流对象的 `readLine()` 方法来读取来自客户端输出的同步数据。实现了服务器端的接收数据功能之后, 就需要对所接收到的数据进行合法性判断。如果接收到的同步数据是不合法的就将该丢弃该同步数据并且不做任何操作, 如果该同步数据合法就需要将该合法的同步数据发送给客户端。因此, 服务器端为了能够将同步数据发送给客户端, 服务器端需要使用 `Socket` 对象的 `getOutputStream()` 方法获取输出流对象, 然后使用输出流对象的 `println()` 将同步数据发送出去。实现上述功能之后, 客户端就可以从服务器端读取到同步数据了。

服务器端功能实现是在集成开发环境 Eclipse 中使用 Java 编程语言编写的, 其工程目录如图 6 所示。

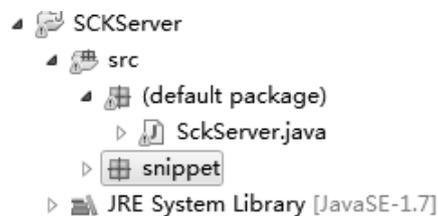


图 6 服务器端工程目录

5 系统测试

测试系统能否正常工作分为以下几步^[7,8]:

① 将 TQ210 开发板和电话模块连接起来, 组成系统测试所需要的搭载了 Android 系统的有线电话, 并打开开发板电源。

② 打开 Eclipse 集成开发环境, 首先启动服务器端的程序, 使服务器处于工作状态。然后, 在手机端和有线电话端安装通讯录的 apk, 并且使手机端和有线电话端的通讯录都处于工作状态。

③ 测试联系人模块的实时同步功能, 首先, 进入手机端和有线电话端通讯录的联系人操作界面。然后, 在手机端新增一个联系人保存该联系人之后, 此时可以看到手机端和有线电话端同时新增了同一个联系人, 如图 7 所示。



图 7 添加联系人同步结果

④ 在手机端通讯录编辑刚才新添加的联系人, 将该联系人的电话号码进行修改, 保存之后可以观察到手机端的联系人和有线电话端的联系人的电话号码同时发生了改变。如图 8 所示。



图 8 编辑联系人同步结果

⑤ 在手机端通讯录里把刚才的联系人信息进行删除, 删除之后可以同时看到有线电话端的同一个联系人信息被删除了。如图 9 所示。



图 9 删除联系人同步结果

⑥ 手机和有线电话同时进入通讯录的通话记录界面, 然后向手机拨打电话, 通话结束后可以看到手机和有线电话的通话记录同时更新了同一条信息, 可以通过观察通话记录上面的时间确定通话记录的实时同步。如图 10 所示。



图 10 通话记录同步结果

经过上面的测试结果可以知道, 该通讯录实现了通话记录和增加、修改和删除联系人的实时同步功能。

6 结语

经过上面的测试, 可以得出结论: 该通讯录具有在通话记录和添加、删除、修改联系人的实时同步功能。事实上, 可以把这个实时同步功能应用到办公人员的手机和办公电话上。当办公人员在外工作时, 如果办公电话有未接来电时, 办公人员可以即时知道这条来电信息并及时回复。也可以将该通讯录应用到具有多部手机的用户的手机上, 只要每部手机都安装了该通讯录, 就可以实现多部手机的通讯录数据的实时同步, 这样可以免去用户在多部手机的同一个联系人进行相同的添加、删除和修改的操作。再则, 随着 4G 和 5G 的发展, 手机在室内的通话质量越来越差, 在室内使用有线电话通话会比手机好, 所以实现这个实时同步功能可以方便用户在有线电话上拨打手机端的联系人, 免去查询电话号码的麻烦。因此, 这个通讯录的实时同步功能具有很好应用前景。

参考文献

- 1 薛莹, 徐慨, 黄麟舒. 来电显示电路的设计. 舰船电子工程, 2008, 28(9): 44-47.
- 2 马庆钟, 姜弢. 基于 SyncML 的 CooTalk 通讯录同步设计与实现[硕士学位论文]. 哈尔滨: 哈尔滨工程大学, 2013.
- 3 李辉, 杨若瑜. 基于安卓手机的信息云同步系统的设计与实现[硕士学位论文]. 南京: 南京大学, 2014.
- 4 熊积健, 王琪. 基于 S5PV210 平板电脑的设计. 江西通信科技, 2012, (1): 12-15.
- 5 姚昱旻, 刘卫国. Android 的架构与应用开发研究. 计算机系统应用, 2008, 17(11): 110-112.
- 6 凡威, 周渊平. 基于 Android 平台的无线视频监控. 计算机系统应用, 2015, 24(1): 195-198.
- 7 杜江, 周渊平. 基于 Android 的电话拨号功能. 计算机系统应用, 2014, 23(12): 245-248.
- 8 陈成伟, 周渊平. 基于 Android 的有线电话 CID 功能. 计算机系统应用, 2016, 25(1): 85-89.