

北京邮电大学

本科毕业设计(论文)



题目: 基于移动互联网的流媒体点播系统的设计与实现

姓 名 许金良

学 院 电子工程学院

专 业 电子信息科学与技术

班 级 2010211204

学 号 10210929

班内序号 05

指导老师 王尚广副教授

2014 年 6 月

基于移动互联网的流媒体点播系统的设计与实现

摘 要

随着第四代数字通信技术(4G)的广泛使用,移动网络带宽大幅度提高;与此同时,以智能手机为代表的移动终端价格日益下降、性能不断提高。以此为契机,以视频点播为代表的移动多媒体业务,由于其相对于传统移动网络业务的优势必将大放光彩,必将在很短的时间内赢得行业和用户的认同。

在此背景下,本文设计和实现了一个基于 RTSP 协议的流媒体点播系统,包括 Android 客户端、流媒体服务器、socket 服务器和数据库服务器,实现了用户注册、登录认证、节目选择、视频播放、查询视频信息等功能。本系统经实验室局域网测试,系统能正常、可靠地运转。本文的主要工作包括:

- (1) 分析了智能手机的发展前景,研究了流媒体的应用现状和技术,并通过对市场主流智能手机操作系统的进行比较,确定使用 android 操作系统作为本文的开发平台。阐述了 android 系统的概况,并对 Android 的多媒体架构进行了讨论。
- (2) 在开源软件 DSS (Darwin Streaming Server) 的基础上搭建流媒体服务器。
- (3) 编写了基于 socket 的控制管理器,作为流媒体系统控制的服务器。
- (4) 建立用户信息和视频信息的数据库,为系统提供信息支持。
- (5) 基于 Android API(Android 应用程序开发接口)和 JNI 开发一款多功能、多个 UI 控制的流媒体客户端,负责用户注册、登录认证、节目选择、视频播放、查询视频信息时与用户的交互。

关键词 流媒体 RTSP socket Android JNI

Design and implementation of mobile Internet-based streaming media on-demand system

ABSTRACT

With the fourth generation of digital communications technology (4G) is widely used ,mobile network bandwidth is greatly improved; At the same time, mobile terminals represented by smart phones is becoming cheaper and cheaper, and performance of which improving continuously. As an opportunity , mobile multimedia services, represented by VOD(video on demand) due to their advantages compared to traditional mobile network business is bound to shine, and will win industry and user identity in a very short time.

In this context, this paper designed and implemented a system based on demand streaming RTSP protocol, including the Android client, streaming media server, socket server and database server, giving users the access to user registration, login authentication. The system has been tested in LAN of our laboratory, the system can reliably operate. The main work includes:

- (1) Analysis of the prospects for the development of smart phones, to study the situation and the application of streaming media technology, and through the mainstream smartphone operating systems are compared to determine the use of android operating system as a development platform for this article. Describes the overview of android system, and Android multimedia framework were discussed.
- (2) Build a streaming media server based on open source software DSS (Darwin Streaming Server) on.
- (3) Write a server socket based control manager, as streaming media system control.
- (4) Establish a database of user information and video information, the system provides information support.
- (5) Based on Android API (Android application development interface), and JAVA development of a multi-purpose, multiple streaming media client UI control, responsible for user registration, login authentication, program selection, video player, check the video information when interacting with the user.

KEY WORDS Streaming RTSP socket Android JN

目 录

第一章 绪论.....	1
1.1 研究背景.....	1
1.1.1 移动互联网的发展状况.....	1
1.1.2 Android 生态系统的概述.....	2
1.1.3 移动流媒体国内外的的发展.....	4
1.2 主要工作.....	5
1.3 论文结构.....	6
第二章 相关技术.....	7
2.1 Android 系统.....	7
2.1.1 Android 系统架构.....	7
2.1.2 Android 多媒体整体架构.....	9
2.1.3 SDK&JNI 技术.....	13
2.2 流媒体服务器.....	15
2.2.1 流媒体概述.....	15
2.2.2 实时流媒体协议 RTSP/RTP/RTCP.....	16
2.2.3 流媒体系统的工作方式.....	16
2.2.4 三大流媒体服务器平台.....	17
2.3 基于 socket 的通信机制.....	19
2.4 POSIX 规范的 MySQL 开发的 C 接口.....	21
2.4.1 数据结构.....	21
2.4.2 相关操作函数.....	22
第三章 需求分析.....	23
3.1 需要具有的功能.....	23
3.2 用户注册和登录	24
3.3 视频的选择.....	25
3.4 基于 Android 的流媒体播放器.....	26
3.5 查看当前视频信息.....	26
第四章 开发环境与平台搭建.....	27
4.1 开发及运行环境.....	27
4.1.1 硬件平台.....	27

4.1.2 开发工具.....	27
4.2 Android 系统开发的平台搭建.....	27
第五章 系统概要设计.....	31
5.1 系统总体设计.....	31
5.2 数据库服务器的设计.....	31
5.3 流媒体服务器的设计.....	32
5.4 移动客户端的设计.....	32
5.4.1 模块划分.....	32
5.4.2 UI 界面设计.....	33
5.5 移动客户端和 socket 服务器的规约.....	34
第六章 系统详细设计和实现.....	37
6.1Android 手机客户端的实现.....	37
6.1.1Android 客户端工程类图.....	37
6.1.2 各个类内部实现的解析.....	38
6.2Socket 服务器端的开发.....	45
6.3 移动客户端和 socket 服务器端的协议.....	47
6.3.1 移动客户端向 socket 服务器端发送的请求指令的格式.....	47
6.3.2 socket 服务器端向移动客户端发送的应答数据的格式.....	47
6.3.3 协议的具体内容.....	48
6.4 流媒体服务器的布置.....	48
6.4.1 安装 DSS 流视频服务器.....	48
6.4.2 部署视频文件.....	49
6.4.3 流媒体服务器的测试.....	49
6.5 数据库的设计.....	50
6.5.1 视频信息数据表(streamMovieList).....	50
6.5.2 用户信息数据表(userInfoList).....	51
第七章 系统测试.....	53
7.1 Android 客户端的测试.....	53
7.2 socket 服务器端的运行记录.....	55
7.3 数据库端的测试.....	57
7.4 流媒体服务器的运转.....	58
第八章 结束语.....	61
8.1 总结.....	61
8.2 展望与改进方向.....	62

参考文献.....	65
致 谢.....	67

第一章 绪论

1.1 研究背景

1.1.1 移动互联网的发展状况

2013 年中国移动网民规模为 5 亿人，渗透率为 81%，其中 3G 用户规模为 4.17 亿户，三大运营商的 3G 用户规模均过亿，较去年均有不同程度的增长。中国智能手机出货量为 3.18 亿台，同比增长 64.1%，随着用户基数的扩大，预计未来几年出货量规模增速将逐渐放缓，保持平稳发展态势。中国移动互联网的繁荣从反映了未来几年移动互联网在世界范围内将保持如火如荼的发展态势¹。

首当其冲的移动互联网超越 PC 互联网，引领发展新潮流。有线互联网(又称 PC 互联网、桌面互联网、传统互联网)是互联网的早期形态，移动互联网(无线互联网)是互联网的未来。手机作为人们生活的必备品，已经在不知不觉中把很多东西从我们的生活中脱离出去，比如 mp3、手表、照相机等。PC 机只是互联网的终端之一，智能手机、平板电脑、电子阅读器(电子书)已远远超过电脑的数量，电视机、车载设备正在成为终端，冰箱、微波炉、抽油烟机、照相机，甚至眼镜、手表等穿戴之物，都可能成为泛终端。

其次就是移动互联网和传统行业的融合呈现新的特点。在移动互联网、云计算、物联网等新技术的推动下，传统行业与互联网融合的平台和模式都发生了改变。这一方面可以作为业务推广的一种手段，如食品、餐饮、娱乐、航空、汽车、金融、家电等传统行业的 APP 和企业推广平台，另一方面也重构了移动端的业务模式，如医疗、教育、旅游、交通、传媒等领域的业务改造。

再次是移动互联网商业模式多样化。随着移动互联网的快速发展，企业不断开辟着新的盈利模式。微信平台开启了移动社交成功的典范，在其之后，易信、陌陌等一定社交平台如雨后春笋般出现。针对不同人群，新兴的移动游戏点燃了市场，像火爆一时的“水果忍者”、“神庙大逃亡”等，移动游戏毫无疑问也是未来移动互联强有力的竞争市场。而前面提到的 4G 时代的到来，宽带的传输速度以及流量得到改善和提高的同时，也催生了移动视频市场。随着移动互联网发展进入快车道，网络、终端、用户等方面已经打好了坚实的基础，不盈利的情况已开始改变，移动互联网已融入主流生活与商业社会，货币化浪潮即将到来。移动游戏、移动广告、移动电子商务、移动视频等业务模式流量变现能力快速提升。

最后是云计算的这种模式的普及，使网络资源得到最大化的运用。而这种网络化资源一定是全球性的。个人的配置再高、硬盘再大，计算能力和存储量始终是有限的。有了云计算这种模式，全世界的资源就可以为我所用，而且这些信息

¹ <http://report.iresearch.cn/2145.html>

不在我的个人设备里。云计算使得全球性的网络平台变成个人的计算平台、分享模式。所以，云计算远远超过了单个设备的能力，使所有的扩展性资源实现按需服务。云计算是共享性的，加上在信息技术里的开放性意识

移动互联网迅速崛起发展，移动互联网应用到生活的方方面面，移动互联网的服务质量问题也开始引起了大家的关注，特别是对网络质量要求高的应用，如网络音视频、网络电话和对网络安全要求比较高的方面。

1.1.2 Android 生态系统的概述

(1) Android 的发展历程

Android 平台是谷歌公司在 07 年末发布的，这是一款开源的、基于 Linux 平台的开源操作系统，旨在打造一款专门服务于移动端的开放、自由的移动平台。主要由操作系统、中间件、UI 和包罗万象的应用构成。

Android 平台是谷歌公司在移动互联网方面的核心产品，以“随时随地为每个人提供信息”为终极目标。通过和上下游开发商、制造商及运营商合作，建立标准化、开放式的移动电话软件平台，在移动通信领域发挥影响力。尤其是开放手机联盟(Open Handset Alliance)的成立，具有标志性意义，将极大地促进这一进程的实现。Android 将继续推进移动设备的不断更新，设计出更加友好的人机交互界面并开发出更加优秀的应用服务。

Google Play 前名为 Android Market，是一个由 Google 为 Android 设备开发的在线应用程序商店。一个名为“Play Store”的应用程序会预载在允许使用 Google Play 的手机上，可以让用户去浏览、下载及购买在 Google Play 上的第三方应用程序。2012 年 3 月 7 日，Android Market 服务与 Google Music、Google 图书、Google Play Movie 集成，并将其更名为 Google Play。Android Market 于 2008 年 10 月正式上线，开始只允许发布免费 APP，现在已开发付费 APP 发布与下载，这让个体开发人员的热情打打提高，令 Android 平台的发展速度大大提升。

自 2008 年推出第一款基于 Android 的手机产品以来，Android 平台发展迅速。至今已经发布 Android4.4 系统。各版本系统都在手机市场有着广泛的应用，目前 Android 系统的智能手机越来越受到人们的推崇。

(2) Android 的特征

Android 的飞速发展与其良好的特征密切相关，Android 系统具有很多特征，主要特征如下。

开放性：Android 平台是开源的免费平台。Google 通过和开发商设备制造商和运营商等机构的不断深入合作，也逐步将 Android 推广为一个立体的、开放式的生态系统。

应用程序无界限：Android 上的应用程序可以通过标准 API 访问核心移动设

备功能。通过互联网，应用程序可以声明它们的功能可以供其他应用程序使用。

应用程序可以并行运行：Android 是一种完整的多任务环境，其中应用程序可以并行运行。Android 平台提供了 4 种程序间的通信方式，解决了一直以来的程序间的交互的难题。

应用程序可以轻松地嵌入网络：应用程序可以轻松地嵌入 HTML、Javascript 和样式表，应用程序可以通过 WebView 显示网络内容；而且 Javascript 可以与 Java 实现无缝衔接。

可配置：Android 上所有的应用程序都是可以替换或扩展的，包括诸如 HOME、Contacts 类核心组件。开发人员完全可以勾勒出属于自己的 Android 蓝图。

可选的虚拟键盘：Android 系统同时支持这物理键盘和虚拟键盘，极大地丰富了用户的输入选择。当前，虚拟键盘已经成为 Android 智能终端的主流输入方式。

Android 的特点还有很多，正是这些优势让 Android 平台得到飞速的发展，Android 的未来令人充满期待。

(3) Android 的市场份额

从处于劣势到得到无数制造商的支持，如今几乎所有的设备厂家（除了 Apple 和 Nokia）都推出了 Android 设备。Android 短短几年不仅彻底打垮了如日中天的 Symbian，还抢占了移动终端超过一半的市场份额，日渐威胁着 IOS 生态系统。苹果在低端智能手机市场上产品有限，部分市场份额遭 Android 蚕食。而 Windows Phone 增长完全归功于诺基亚及在欧洲、亚洲和美国不断改进的 Lumia 组合，可以说微软在全球的移动市场份额仍处于一个较低水平。

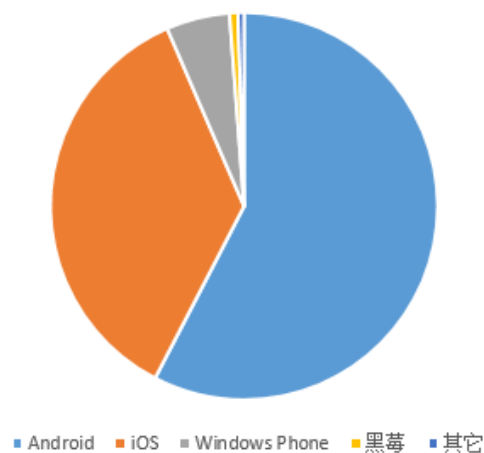


图 1-1 美 2014Q1 移动市场份额

市场调研公司 Kantar Worldpanel Comtech 发布的统计数据显示，Android 系统 2014 年第一季度在美市场收获颇丰，同比增长 8.3%至 57.6%；iOS 第一季度在美市场份额下滑了 7.8%至 35.9%；Windows Phone 第一季度在美市场份额为 5.3%，排名第三，黑莓排名第四，市场份额为 0.7%。

1.1.3 移动流媒体国内外的的发展

流媒体技术是指对时间连续的音视频等多媒体文件进行压缩编码处理，然后存放在流媒体服务器中，用户在客户端可以采用一边下载一边播放的方式，从而实现多媒体数据的网络实时传输播放的网络传输技术。传统的网络传输播放方式需要用户将这个多媒体文件完全下载之后才能播放，在这种情况下，用户在完全下载下来之前无法知道多媒体的内容是什么以及是否是其所需要的。不仅如此，用户可能只是仅仅播放一遍多媒体文件，并不打算把它存储在硬盘上。基于这种对比，流媒体技术降低了播放的延迟时间，提高了多媒体的实时性，并且为客户端节约了不必要的存储空间。移动流媒体技术是指在移动互联网络环境下采用流媒体技术实现多媒体数据的网络传输技术。移动流媒体作为新兴的一种流媒体应用领域，随着移动互联网络和移动终端的发展，它的应用前景越来越广阔。

在信息技术领域，多媒体技术应用是发展最快、最为活跃、最贴近大众生活的技术之一，主要涉及到图像、声音、动画、视频等多种媒体的内容，多媒体信息所包含的数据量非常大，对存储设备容量，通信网速度和处理器速度都有很高的要求，所以原生的多媒体数据难以大规模应用。为了解决这个难题，需要采用多媒体数据压缩技术，在文件容量和多媒体质量之间做个折中，在不影响用户体验的情况下尽量减小数据量，使多媒体数据的网络传输成为可能。

多媒体数据之所以能够压缩，是因为图像、声音、视频数据一般具有很大的压缩力。拿位图格式的图像存储方式来说，在这种形式的图像数据中，像素与像素之间在行方向和列方向都具有很大的相关性，因而整体上具有很大的数据冗余；在允许一定限度失真的前提下，能对图像数据进行很大程度的压缩。多媒体数据压缩技术既包括软件技术又包括硬件技术，它们的核心都是对数据执行基于压缩算法的数学运算。音视频的压缩编码技术在近年来得到了不断进步和发展，已经取得了惊人的成果，并得到了工业界的广泛支持 and 应用。国际上制定视频编解码技术的组织有两个，一个是“国际电联（ITU-T）”，它制定的标准有 H. 261、H. 263、H. 263+ 等，另一个是“国际标准化组织（ISO）”它制定的标准有 MPEG-1、MPEG-2、MPEG-4 等。而 H. 264 则是由两个组织联合组建的联合视频组（JVT）共同制定的新数字视频编码标准。H. 264 标准具有较高的复杂度，但是具有更高的编码效率、高质量的视频画面、提高网络适应能力、可以应用在不同场合等优点，因而自发布以来成为了一种被广泛使用的高精度视频的录制、压缩和发布格式的标准。而在音频编码方面，目前主流的编码格式有 MP3、AMR、AAC 等。

随着移动互联网络技术的不断发展和移动智能终端不断地得到普及和发展，移动多媒体技术成为了移动应用的一个新的课题，多媒体数据在移动网络的传输是当前一个研究热点，具有广阔的发展前景。中国移动宣布将建成全球最大 4G 网络，目前正在加快推进网络部署，年底基站总数将超过 50 万，目前中国移动可向北京、上海、广州、深圳等 16 个城市提供 4G 服务；预计年中预计 100 个城

市具备 4G 商用条件；年底，将有超过 340 个城市的客户可享受到中国移动的 4G 服务。去年 12 月，工信部向三大运营商颁发了 TD-LTE 4G 牌照。TD-LTE 是我国主导的国际主流 4G 标准，也是当前全世界最先进的通信技术之一，具有高速率、低时延、永远在线的显著特点。目前，全世界已有 18 个国家共部署了 23 张 TD-LTE 商用网络，另有 40 余个商用网在计划部署中，全球用户规模已突破 500 万，形成了具有相当规模的运营商和厂家合作平台，TD-LTE 产业链已经初具规模。与此同时，移动智能终端的处理能力和硬件性能较之以往的商务手机有了较大的提高，这就为移动流媒体的应用提供了硬件基础。

虽然移动网络已经得到了很大的发展，但是移动网络的多媒体传输仍然受到了很大的局限。首先，移动网络的带宽仍然有限，提供多媒体服务特别是视频服务还很吃力。其次，移动网络的丢包率和误码率比较高，网络状况较为不稳定，需要对多媒体的传输进行传输控制，提高传输的质量。最后，移动终端的处理能力相对于 PC 电脑低了很多，需要根据移动终端的特点，对移动流媒体系统的设计和实现进行相应的优化，才能使移动流媒体成为可能。

1.2 主要工作

设计并实现了一套基于移动互联网的流媒体点播系统，包括 Android 客户端、流媒体服务器、socket 服务器和数据库。

Android 客户端用 Eclipse 开发，并使用了 NDK 开发工具，编程语言包括 Java 语言和 C 语言。用户在客户端上能够完成用户注册、用户登录、节目选择、视频播放、播放控制、查询视频信息等功能。

流媒体服务器采用了 Darwin Streaming Server，布置在 Ubuntu 平台上。它能够响应用户请求，将响应的视频文件流花，并向用户分发基于 RTP 协议的包含视频数据的报文分组。

Socket 服务器相当于 Android 客户端和数据库之间的转发控制器。它响应用户请求，并对用户请求进行解析，然后根据用户的请求指令对数据库进行操作，并把状态信息和结果数据返回给用户。

数据库采用了 MySQL，共设计了两张数据表，分别是视频信息数据表 and 用户信息数据表。视频信息数据表，用来存储视频信息，如视频名称、URL、作者和详细信息；用户信息数据表，用来储存用户名、密码、昵称和邮箱。

通过在局域网中对系统进行测试，发现系统运行良好，具有响应快捷、视频清晰流畅、控制方便、界面人性化的优点，并支持并发访问，很好地实现了应有的功能。

1.3 论文结构

第一章：绪论。首先描述了论文所研究问题的应用背景和意义，以及国内外移动流媒体技术及应用的发展状况，然后简单介绍论文的主要工作和章节安排。

第二章：相关技术。简要介绍了系统实现的技术基础，主要讲述了一些技术概念，为后续的章节提供技术环境和基础。本章的内容安排依次是 Android 系统，包括 Android 系统架构、多媒体整体架构和 NDK&JNI 技术概述；流媒体服务器，包括流媒体的概念、流媒体协议簇、流媒体系统工作方式和三大流媒体系统概述；socket 通信机制，简述了 socket 客户端和服务器的交互机制和通信原理；MySQL 数据库开发基础，简述了 C 语言开发 MySQL 数据库所用到的一些数据结构和操作函数。

第三章：需求分析。本章从用户角度简述了系统开发的限制条件，包括系统运行的环境和开发平台、系统用例图，本章的内容为后面章节的具体开发提供了依据。

第四章：系统概要设计。本章的内容基于第二章，不过看待问题的视角是从开发者的角度。本章概述了系统各个模块的功能和模块之间的规约。

第五章：系统设计和实现。对第三章各模块的功能进行深入探究，分析各个类的功能划分及其之间的关系，并实现各模块的功能。

第六章：系统测试分析。对系统的各个功能进行测试。

第七章：结束语。对本文的工作进行总结，探究其不足之处并提出改进方向。

第二章 相关技术

2.1 Android 系统

2.1.1 Android 系统架构

Android 系统从上到下分为 4 层：Linux 内核、Android 核心库及 Android 运行时环境、应用程序框架层和应用程序层，如图 2-1 所示：

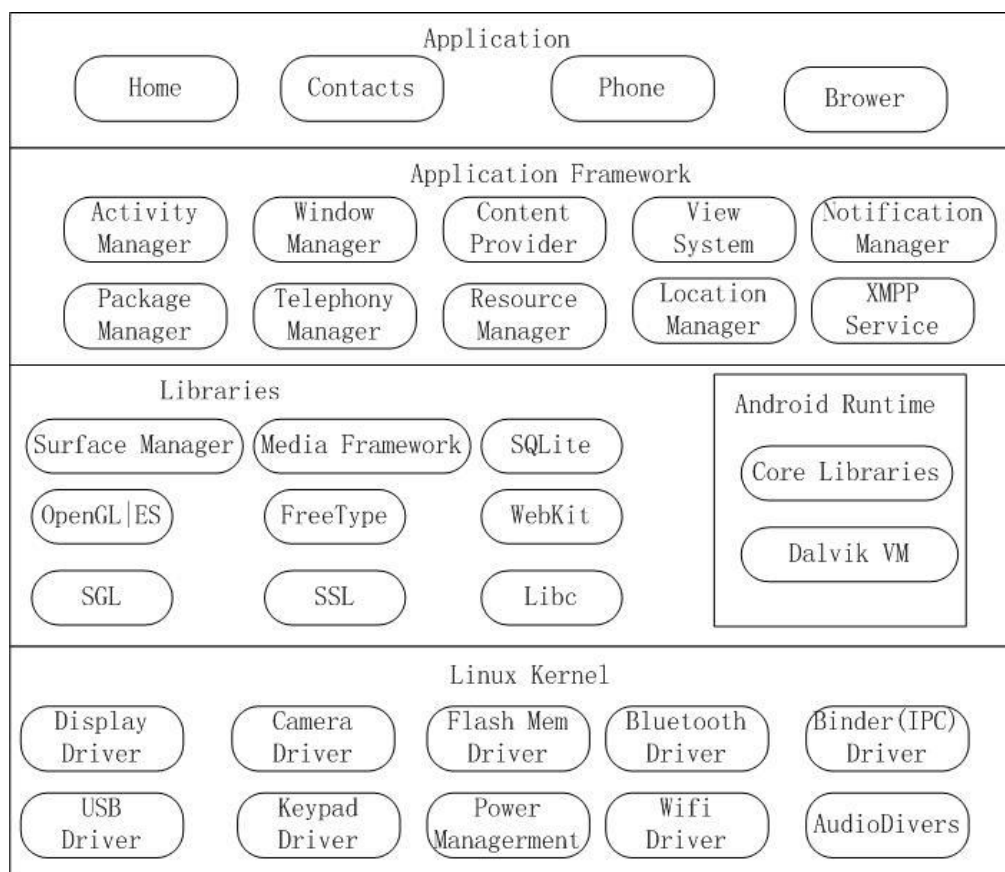


图 2-1 Android 系统架构

这样分层的好处是能够调用下层提供的服务并为上层提供统一的服务接口，本层及下层的具体实现对于上层是透明的，当本层及下层的具体实现发生变化时不会对上层造成影响。这样的部署有利于形成高内聚、低耦合的良好架构。

(1) Linux 内核层(Linux kernel)

Android 对操作系统的使用包括两个部分，核心和驱动程序两个部分，Android 的 Linux 核心是标准的 Linux 2.6 内核，Android 更多的是需要一些与设备相关的驱动程序。主要的驱动程序如下：

- ① 显示驱动：基于 Linux 的帧缓冲驱动。
- ② Flash 内存驱动：基于 MTD 的 Flash 驱动程序。
- ③ WiFi 驱动：基于 IEEE 802.11 标准的驱动程序。

- ④ 键盘驱动：作为输入设备的键盘驱动。
- ⑤ 蓝牙驱动：基于 IEEE 802.15.1 标准的无线传输技术。
- ⑥ Binder IPC 驱动：Android 的一个特殊的驱动程序，具有单独的设备节点，提供进程间通信功能。
- ⑦ 电源管理：管理电池电量等信息。

(2) Android 核心库(Libraries)

Android 包含一个 C/C++库的集合，以供 Android 系统的各个组件使用。这些功能通过 Android 的应用框架(Application Framework)暴露给开发者。

下面是一些核心库：

①系统 C 库：C 语言的标准库，也是系统中一个最底层的库，C 库是通过 Linux 的系统调用实现的。

②媒体库：Android froyo 版本多媒体引擎做了变动，新添加了 stagefright 框架，并且默认情况 android 选择 stagefright，弃用之前的 opencore，仅仅对 opencore 中的 omx-component 部分做了引用。多媒体库是 Android 多媒体的核心部分，从功能上一共分为两大部分，其中一部分是音频、视频的播放；另一部分是音频、视频的录制。

③界面管理：显示子系统的管理器，管理访问显示子系统和无缝组合多个应用程序的二维和三维图形层。

④SGL:基本的 2D 图像引擎。

⑤OpenGL ES：该库使用硬件 3D 加速或使用高度优化的 3D 软加速。

⑥FreeType：位图(Bitmap)和矢量(Vector)字体的功能。

⑦SQLite:一个通用的嵌入式数据库，提供强大而轻量级的关系数据库引擎。

(3) Android 运行时环境(Android Runtime)

该层包括 Dalvik 虚拟机及 Java 核心库，提供了 Java 编程语言核心库的大多数功能。

Dalvik 虚拟机是 Android 系统使用的 Java 虚拟机。每一个 Android 应用程序都有一个虚拟机实例，运行在各自的进程中。Dalvik 虚拟机可执行文件的格式是 .dex，这是一种专为 Dalvik 设计的压缩格式，适合内存和处理器速度都有限制的系统。

大多数虚拟机包括 JVM 都是基于栈的，而 Dalvik 虚拟机是基于寄存器的。一般来说，基于栈的及其需要更多指令，而基于寄存器的机器指令更大。Dx 是一套工具，可以把 java.class 转换成 .dex 格式。

Dalvik 虚拟机依赖于 Linux 内核提供的基本功能，如线程和内存管理。

(4) 应用程序框架层(Application Framework)

应用程序框架层为应用程序开发者提供 API，这样使得组件的重用得以简化，任何应用程序都能够发布它的功能且任何其它应用程序都可以使用这些功能。这种机制使得开发者可以编制极其丰富和新颖的应用程序，自由地利用设备硬件优

势、访问位置信息、运行后台服务、设置闹钟等。

每个应用程序其实就是一组服务框架，包括如下几项：

①视图(View):丰富的、可扩展的视图集合,用来构建应用程序。包括列表、网格、文本框、按钮等,甚至是可嵌入的网页浏览器。

②内容提供者(Content Providers):使应用程序可以访问其它应用程序(如通讯录)的数据,或共享自己的数据。

③资源管理器(Resource Manager):提供对于非代码资源的访问,如本地化字符串、图形和布局文件。

④通知管理器(Notication Manager):使得应用程序能够在状态栏显示自定义的提示信息。;

⑤活动管理器(Activity Manager)管理应用程序生命周期,并提供常用的导航回退功能。

(5) 应用程序层(Application)

该层是 Java 的应用程序层,Android 内置的 Google Maps、E-mail、即时通信工具、浏览器、MP3 播放器等处于该层,Java 开发人员开发的程序也处于该层,而且和内置的应用程序具有平等的位置,可以调用内置的应用程序,也可以替换内置的应用程序。

2.1.2 Android 多媒体整体架构

(1) 整体架构概述

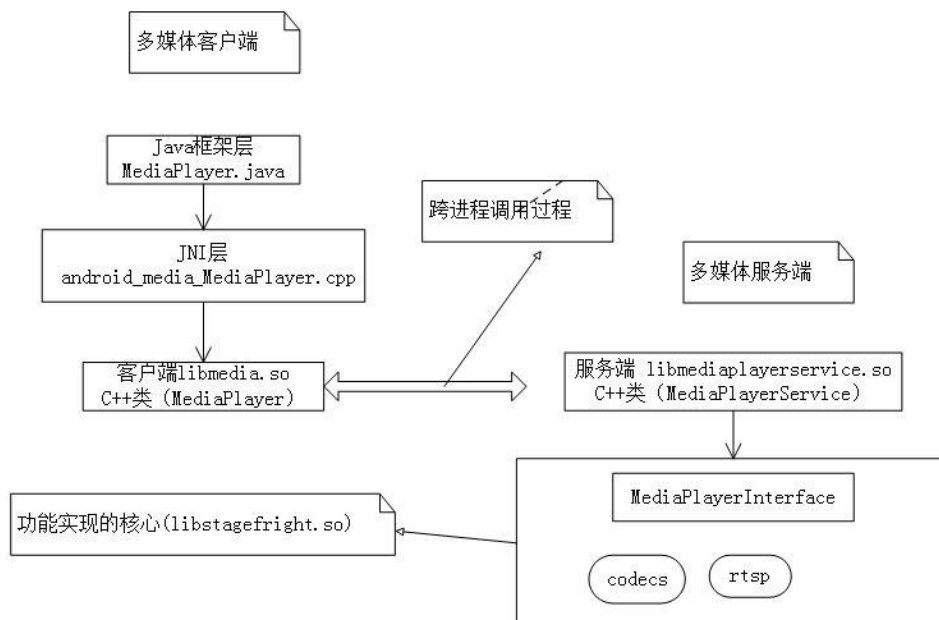


图 2-2 多媒体体系架构

Android 多媒体整体架构如图 2-2 所示,在图示的各个库中,libmedia.so 位于核心的位置,它对上层的提供的接口主要是 MediaPlayer 类,类

libmedia_jni.so 通过调用 MediaPlayer 类提供对 JAVA 的接口，并且实现了 android.media.MediaPlayer 类^[2]。

libmediaplayerservice.so 是 Media 的服务器，它通过继承 libmedia.so 的类实现服务器的功能，而 libmedia.so 中的另外一部分内容则通过进程间通讯和 libmediaplayerservice.so 进行通讯。

libmediaplayerservice.so 的真正功能通过调用多媒体引擎 Stagefright 来完成。

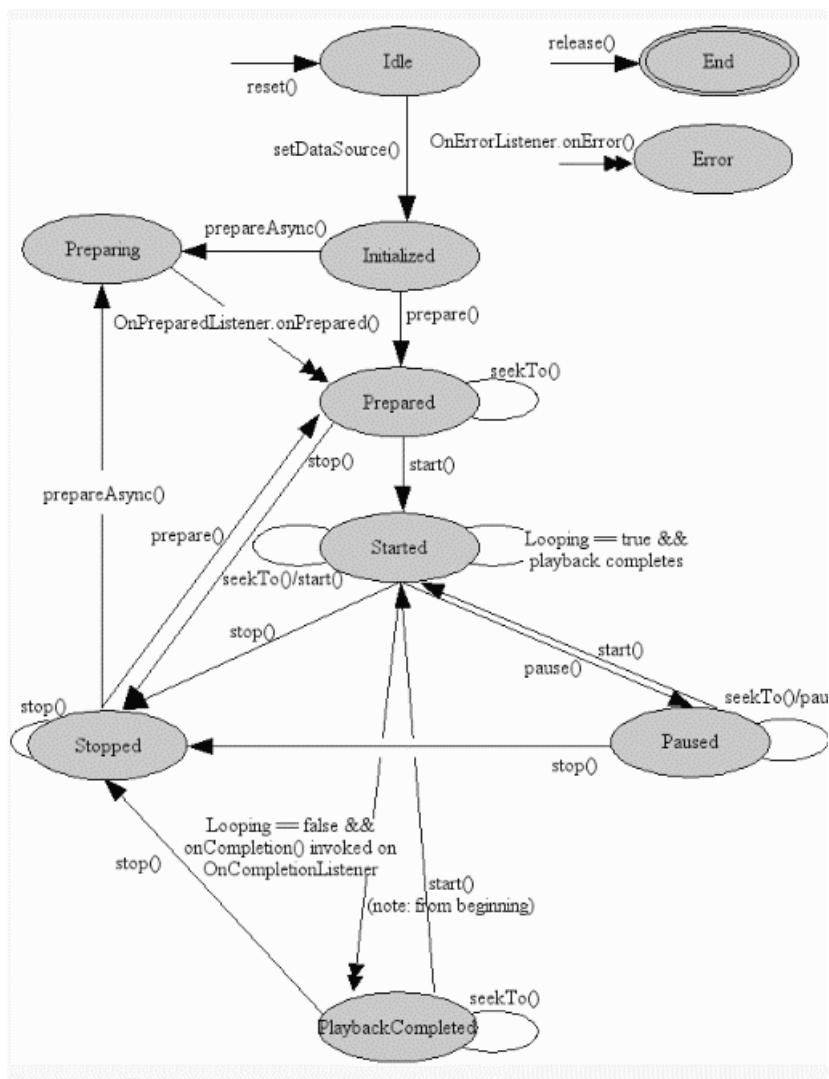


图 2-3 MediaPlayer 对象生命周期

(2) Java 层的 MediaPlayer 类

MediaPlayer 类是播放媒体文件最为广泛使用的类。MediaPlayer 已设计用来播放大容量的音频文件以及同样可支持播放操作（停止、开始、暂停等）和查找操作的流媒体。其还可支持与媒体操作相关的监听器。通过以下方式可完成播放 MediaPlayer 中的音频和视频：

- ① 从源文件播放
- ② 从文件系统播放

③ 从流媒体播放

当一个 MediaPlayer 对象被 new 初始化，或者调用 reset() 时处于 Idle 状态，调用 release() 后处于 End 状态，在这两种状态之间则处于 MediaPlayer 对象的生命周期²。MediaPlayer 对象的生命周期如图 2-3 所示。

(3) StageFright 多媒体引擎

从 Android 2.2 版本开始，Android 对多媒体框架进行了很大调整，抛弃了原来的 OpenCore 框架，而改用了 StageFright 框架，StageFright 仅仅对 OpenCore 中的 omx-component 部分做了引用，这主要是为了录像和视频通话功能，另外在混频和多摄像头支持方面也做了加强。StageFright 框架比 OpenCore 框架更加易懂，封装也相对简单。但是起初 StageFright 框架也有自己的缺点，即推出时间不长，可支持的文件格式较少。随着 StageFright 框架的不断完善，目前 Android 4.4 中 StageFright 框架已支持常见的大部分视频格式。

Stagefright 在 Android 中是以 shared library 的形式存在 (libstagefright.so)，其中的 module -- AwesomePlayer 用来播放 video/audio。AwesomePlayer 提供许多 API，可以让上层的应用程序 (Java/JNI) 来调用。

一个完整的多媒体处理流程由 DataSource、MediaSource、MediaExtractor 等子类节点构成^[2]，下面按顺序依次讲解：

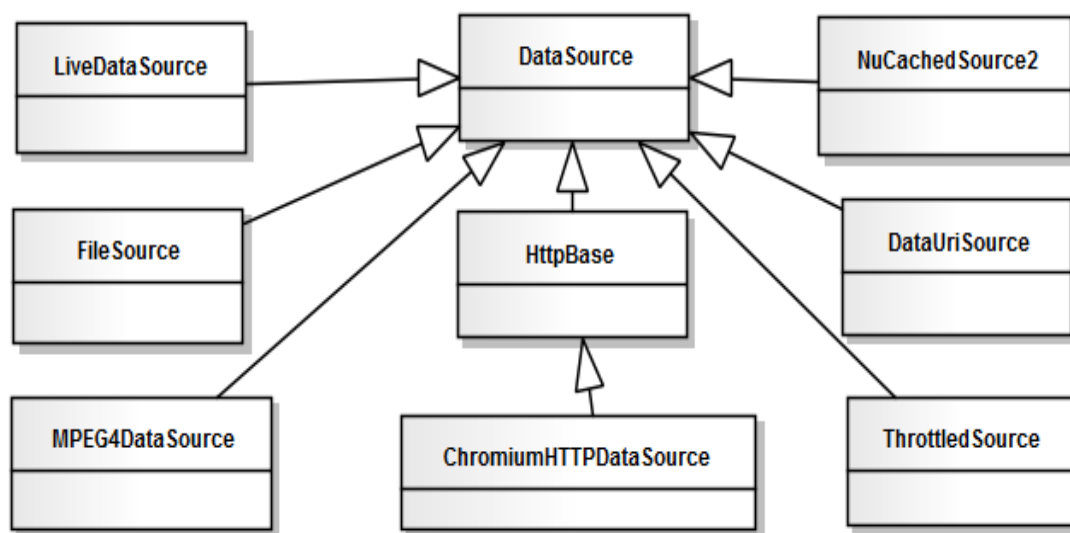


图 2-4 DataSource 的子类节点图

① DataSource 遍历上一步骤所获取的多媒体数据流，分离出音频或视频流数据，它的子类如图 2-4 所示。

² <http://developer.android.com/reference/android/media/MediaPlayer.html>

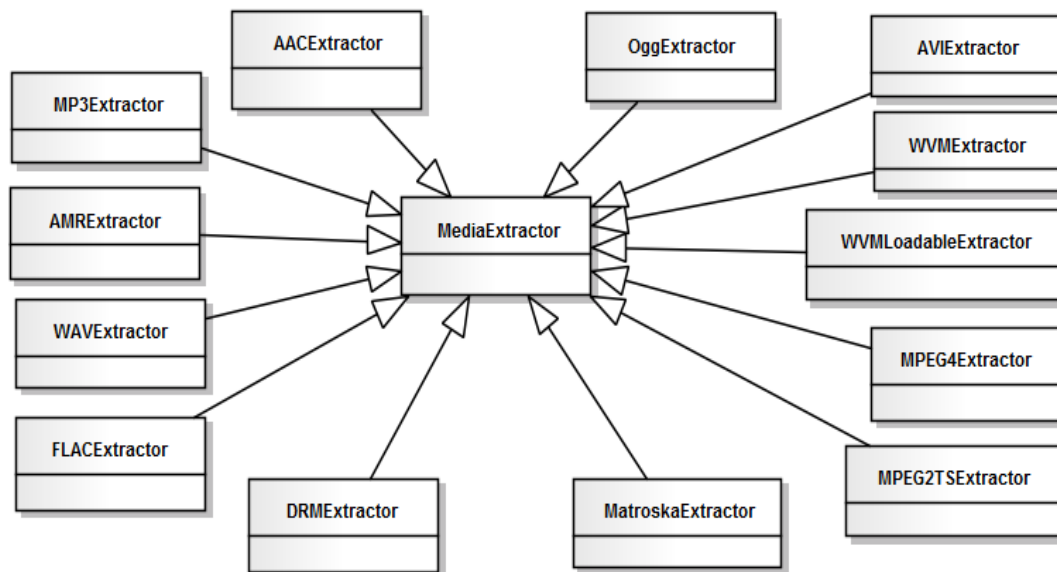


图 2-5 MediaExtractor 的子类节点图

②MediaExtractor 对上步骤分离出来的音频和视频数据进行分析，分别确定音频和视频编码格式。并创建响应的编解码器，对数据进行解码。其子类如图 2-5 所示。

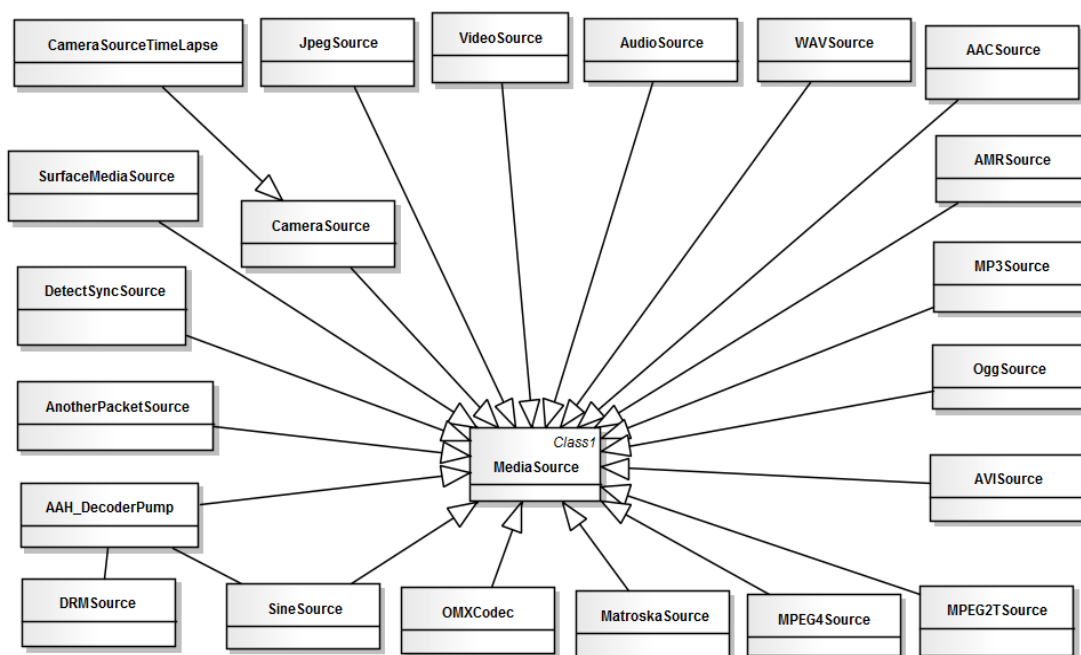


图 2-6 MediaSource 子类节点图

④ MediaSource 遍历上一步骤所获取的多媒体数据流，分离出音频或视频流数据，它的子类如图 2-6 所示。

① VideoRender 类用解码器（OMX Codec）的解码输出的视频数据来渲染视频帧，展示视频的画面。如图 2-7 所示。

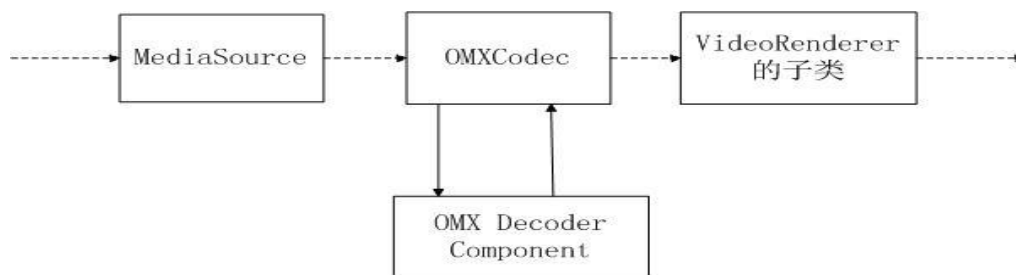


图 2-7 视频渲染结构图

② AudioPlayer 类用解码器 (OMX Codec) 的解码输出的音频数据播放声音。

如图 2-8 所示。

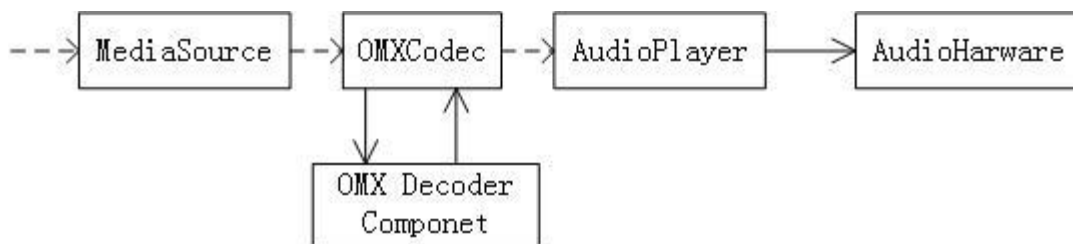


图 2-8 音频数据流图

2.1.3 NDK&JNI 技术

(1) NDK&JNI 概述

JNI (Java Native Interface) 即 Java 本地开发接口，也是 C/C++ 与 Java 通信的一个协议，这个协议可以使 Java 代码与 C/C++ 代码之间进行沟通。通过该协议，可以实现 Java 代码与 C/C++ 代码的互相调用。

NDK (Native Development Kit) 即本地开发套件，提供了一系列的工具，帮助开发者快速开发 C/C++ 的运行库，并能自动将 .so 和 Java 应用一起打包成 apk。NDK 同时提供了交叉编译器，并提供了 .mk 格式的编译配置文件，它类似于 Linux 下的 Makefile，这种配置文件隔离了硬件、CPU、平台的差异，实现了自动编译。

(2) NDK&JNI 的优势^[4]

① JNI 扩展了 Java 虚拟机的能力。Java 语言是不能直接操作硬件的 (Java 在实现跨平台操作时，将硬件操作全部封装起来)，例如，Java 语言不能够直接用来开发驱动，但 Java 可以通过 JNI 来做一些驱动开发 (驱动开发需要操作硬件，而 C 语言可以实现对硬件的操作，Java 通过 JNI 协议，可以实现对 C 的调用)，Java 与 C/C++ 通过 JNI 互调的概要图如图 2-9 所示。

② C/C++ 语言运行效率高。C/C++ 语言的运行效率一般比 Java 要高，应用比较广泛。例如，数学运算 (游戏画面对关照的计算)、实时渲染的游戏 (OpenGL 对显卡的绘制)、音视频处理 (ffmpeg 音视频编解码)。

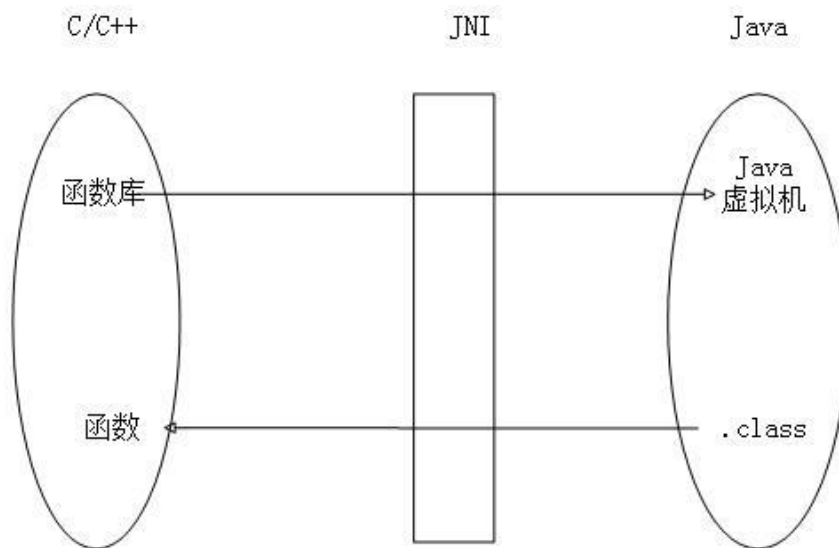


图 2-9 JNI 工作机制

③由于 Java 文件编译成的可执行文件可以很轻易地通过反编译破解其 Java 代码，所以保密性要求高的业务逻辑用 Java 来实现是很不安全的。这种情况下可以讲高性能和高保密性的逻辑封装在 C/C++ 部分。

④直接复用 C/C++ 中的优秀代码。C/C++ 已经过发展多年，开源阵营日益强大，沉淀出了许多优秀的代码项目（如文件压缩、音视频编解码、人脸识别）。这些代码风格优秀，且开发者根据很多用户的反馈不断对其修订，所以功能可靠，效率很高，具有无可比拟的优势。

⑤Java 不可以手动释放内存，而垃圾收集器对内存的释放时机是不可预知的，而 C/C++ 可以实现手动释放内存。如果对内存的利用率有较高的要求时，需要使用 C/C++ 来实现。

⑥特定的应用场景。如高级汽车上安装的车载计算机中的在线故障检测系统，需要获取硬件上的电信号，则此任务非 JNI 莫属。

(3) NDK 与 JNI 的关系

Java 代码通过 JNI 机制和 C/C++ 双向沟通的具体步骤：

- ① 编写包含 native 本地方法的 Java 类。
- ② 借助于 javah 工具生成 C/C++ 的头文件。
- ③ 实现 C/C++ 头文件。

④使用交叉编译工具对 C/C++ 本地代码进行编译，最后连接生成 .so 可执行的 C/C++ 库文件。

⑤实际执行的 Java 代码去和本地的 C/C++ 代码相互沟通。

上述过程是传统的 Java 和 C/C++ 实现交互的步骤，显然这个步骤是繁琐的。

Android 的 NDK 开发工具集是 Android 为了方便开发者通过 JNI 的机制达到 Java 和本地 C/C++ 代码相互沟通的强有力的武器。通过 NDK，Android 开发人

员可以很方便地实现 Java 和本地 C/C++代码的相互调用，充分发挥本地硬件的特性和 C/C++代码的高效性。

2.2 流媒体服务器

2.2.1 流媒体概述

在网络上传输多媒体信息与不包括声音和图像的数据信息有很大的区别，主要有以下两点：

- (1) 多媒体信息的数据量往往很大；
- (2) 多媒体信息的传输对时延和抖动有很高的要求。

由于多媒体数据是分组发送的，而因特网本身是非等时的，所以到达目的地的数据分组就有了延迟和延迟抖动。

图 2-10 很好地阐释了这一点：

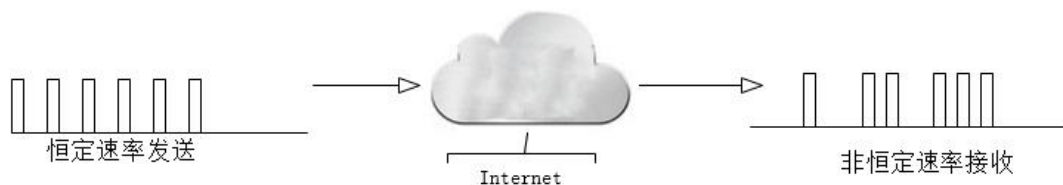


图 2-10 因特网是非等时的

要解决这一问题，可以在接收端设置适当大小的缓存，当缓存的分组数达到一定的数量后再以恒定的速率按顺序将这些分组解码还原并播放。

图 2-11 显示了缓存的作用：

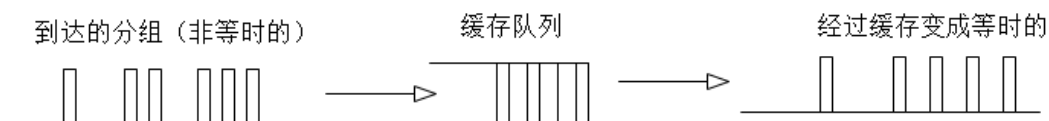


图 2-11 缓存把非等时的分组变换成等时的

上图演示了缓存带来的好处，它能把非等时到达的分组变成非等时的，有利于流媒体的平滑播放。但是，这带来了另一个问题，即时间延迟。从上图可以看到，缓存队列里的分组等待了一段时间之后才能够播放，这客观上增加了时间延迟。因而，从这儿可以看出，时延和时延抖动是流媒体播放系统服务质量的一对矛盾。

目前因特网提供的音频/视频服务大体上可分为三种类型^[5]：

(1) 流式(streaming)存储音频/视频。需要事先把已压缩好的音频/视频文件存储在服务器上，用户可以通过 Internet 边下载边播放。

(2) 流式实况音频/视频。与电视台的实况广播类似，只是要用 Internet 传输数据。音频/视频数据不是事先录制好和储存在服务器中，而是在发送方边录制边发送。

(3) 交互式音频/视频。用户使用 Internet 与其他人进行实时交互式通信。现在的因特网电话或因特网电视会议就属于这种类型。

流媒体就是上面所说的流式存储音频/视频。流媒体的特点是边下载边播放，但不能存储在硬盘上成为用户的文件。

2.2.2 实时流媒体协议 RTSP/RTP/RTCP^[6,7]

基于 UDP 的实时传输协议 (Real-time Transport protocol, RTP) 定义了在网络上传输音/视频的标准数据包格式，它为实时应用提供端到端的运输，但不提供任何服务质量的保证。需要发送的多媒体数据块 (音频/视频) 经过压缩编码处理后，先送给 RTP 封装成 RTP 分组，RTP 分组再装入运输层的 UDP 用户数据包，然后再向下递交给 IP 层。如图 2-12 所示

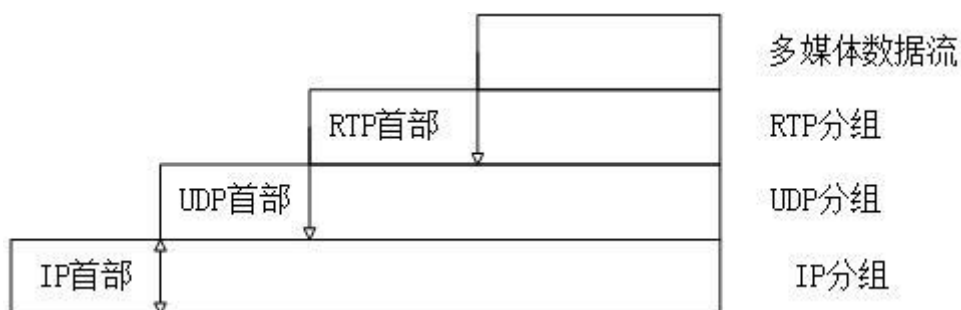


图 2-12 RTP 封装 UDP 数据

实时传输控制协议 (Real-time Transport Control Protocol, RTCP) 和 RTP 一起提供流量控制和拥塞控制服务。在 RTP 会话期间，各参与者周期性地传送 RTCP 包，包中含有传输速率、丢失数据包的数量等统计资料。服务器可以根据这些统计信息动态地改变传输速率，或改变荷载类型。RTCP 有下列四大功能：1、提供视频流的质量反馈，2、通过参与者数量来计算控制包的发送速率，3、带有称作规范名字的 RTP 源持久传输层标识。

基于 TCP 之上的实时流媒体协议 (Real-Time Streaming Protocol, RTSP) 本身并不传输数据，而是作为媒体服务器的网络远程控制，目的是为了在 IP 网络上实现流媒体的有效分发和转发。它定义了具体的控制消息、操作方法和状态吗，可以提供流媒体的暂停、播放、快进等功能。RTSP 既可以应用于大量客户端用户的访问，也可适用只有一个客户端的点播情况。客户端和服务端都可以发出请求和应答。

2.2.3 流媒体系统的工作方式

流媒体系统的工作方式^[5]如图 2-13 所示，主要包括四部分，依次是：

① 万维网服务器：负责解析移动客户端请求、向数据库请求视频元信息并向移动客户端分发视频元信息。其中视频元信息包括视频名称、视频 URL 等，存储

在数据库中。

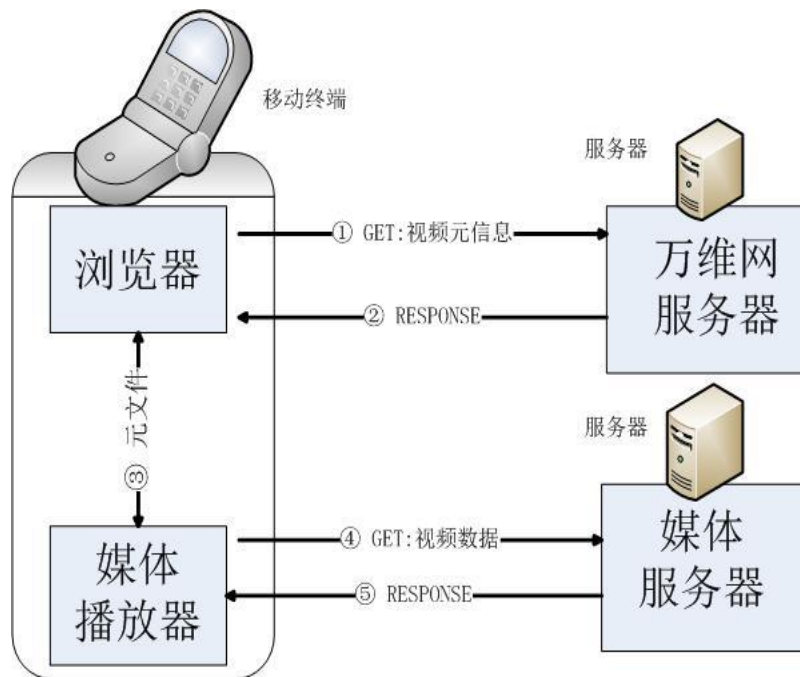


图 2-13 流媒体系统工作方式

②浏览器：向万维网服务器请求视频元信息，并将请求得到的元信息展示给用户，而且能够在用户选定某视频时将该视频的 URL 交给媒体播放器。

③媒体播放器：根据视频的 URL 信息向媒体服务器请求视频数据并负责播放。

④媒体服务器：响应媒体播放器请求并向媒体播放器分发视频数据。

2.2.4 三大流媒体服务器平台

流媒体技术一般都有三个方面的表现，分别是编码器（编码技术）、播放器（播放支持）和流服务器，三者缺一不可。仅仅具有其中一到两个要素是不能被称为流媒体技术的。目前使用较多的流媒体播放技术主要有三种：分别是 RealNetworks 公司的 RealMedia、Microsoft 的 Windows Media 和 Apple 公司的 QuickTime^[8]。

(1) 三大流媒体简介

①RealMedia

首先是 RealNetworks 公司推出 RealMedia 流媒体技术，目前在市场上占据主流地位，拥有最多的网络用户数量。也正是因为出现了 RealMedia，形成了一个个新的行业，如网络广播、网络教学、网络视频点播等。

RealMedia 音频部分使用的是 RealAudio，这种编码能在低带宽环境下实现优质的传输性能，如可以在非常低的带宽（28.8kbps）下提供足够好的音质，所

以 Real System 被认为是在窄带网上最优秀的流式媒体传输系统. 整个 Real System 共分为以下三部分:

媒体制作工具(Real Producer), 用户使用 Real 系列的媒体制作工具可以制作出适合 Real System 在网络上传输的动画、文字、图片等文件内容。与此同时, 通过 Real System 的流式传输技术, 可以将这些媒体文件整合成同步多媒体集成语言, 然后再通过基于 TCP/IP 网络的 Real System 以流式方式来发布, 实现一定的用户交互能力和媒体控制能力, 但与 QuickTime 相比还有一段距离。

服务器端软件(Real Server)可以用于在网络上发布 Real 格式的 Audio、Video 文件, 以流格式在网络上传输播放。如同 Web Server 向网络发布网页, RealServer 对媒体文件是以流式传输的而不是简单的文件下载, 可以在带宽并不充裕的网络上发布, 使成千上万的用户能同时在线观看网络视频。

客户端播放器(Real Player)是一个在网络上对实时传输的音/视频在线收听的工具软件。Real Player 能够播放绝大多数的媒体格式(MP3、AVI、WAV、MOV、RM、RAM、RA)。使用客户端 Real Player, 不仅可以看到实时音/视频流节目, 同时也可以观看点播视频流节目。

②Windows Media

微软推出 Windows Media 的时间比较晚, 依赖其 windows 操作系统抢夺市场份额。Windows Media 对其他音频压缩技术一律不提供直接支持, 更没有公开任何技术细节, 完全封闭。RealMedia 在开放性方面比 Windows Media 就要好一些, 如其实时流协议 RTSP(RealTime Stream Protocol)被采纳为行业标准。

Microsoft 分别采用 MPEG-4 视频压缩技术和自己开发的 Windows Media Audio 音频压缩技术。Windows Media 的关键核心是串流媒体传送协议(Microsoft Media Server protocol, MMS 协议)和高级串流格式(Advanced Stream Format, ASF)数据格式, MMS 用于网络传输控制, ASF 则用于媒体内容和编码方案的打包。微软将 ASF 定义为同步媒体的统一容器文件格式, 音频、视频、图像、控制命令脚本等多媒体信息通过这种 ASF 数据格式, 以网络数据包的形式传输, 实现流式多媒体内容的发布。ASF 是专门为在 IP 网络上传送有同步关系的多媒体数据而设计, 所以 ASF 格式的信息特别适合在 IP 网络上传输。

③QuickTime

Apple 公司的 QuickTime, 不仅仅是一个媒体播放器, 支持 MP4、3GPP、MOV 等文件格式的播放, 而且是一个开放的多媒体架构, 包含了各种各样的流式或者非流式的媒体技术。这种包容性使得它是一个完整的多媒体平台, 可以用它来进行多种媒体的创建、生产和分发, 并为这一过程提供端到端的支持, 在交互性方面是三家之中最好的。在音/视频质量方面, QuickTime 具有跨平台、存储空间要求小等技术特点, 采用有损压缩方式的 MOV 格式文件, 使画面效果较 AVI 格式要稍微好一些。

QuickTime 音/视频流也可以通过诸如 RTP 和 RTSP 等实时协议来分发, 所需

要的流媒体服务器为 Darwin 流媒体服务器。Darwin 流媒体服务器是 Apple 公司开发的开源流媒体服务器架构，可以运行于 Windows、Linux、MacOS 等操作平台上，是目前同类产品中支持操作平台最多的服务器。任何开发人员都可以获取其源码及开发文档，能够从底层研究和了解流式媒体技术，如网络编程、多线程之间的通信、任务调度、系统资源的合理利用、流式标准协议的具体实现等，在此基础上可以对流媒体服务器的功能进行扩展，开发出流式媒体产品，来满足实际的需求。

(3) 三大流媒体服务器比较

表 2-1 三大流媒体服务器对比

名称	RealMedia	Windows Media	QuickTime
开放性	半开放	完全封闭	完全开放
支持并发用户数	3000	2000	2000
交互性和媒体控制能力	弱	中	强
支持的服务器类型	Windows 系列, Solaris, Linux, MacOS	Windows 系列	Windows 系列, Solaris, Linux, Free BSD, MacOS
文件格式	.mp3 、 avi 、 WAV, RM/RMVB	.asf、WMV	.MOV 、 .3GPP 、 MP4
RTSP 协议	支持	不支持	支持

三者之中，微软的 Windows Media 和 Real Networks 的 Real System 是既成的业界标准，但其技术标准对外部不公开，使得开发人员不能完成新功能的开发；而 QuickTime Streaming Server (QTSS, Darwin Streaming Server, DSS) 是完全开源的，任何开发人员都可以对其功能进行扩充甚至修改服务器的核心功能。可以看出 Darwin Streaming Server 对开发者来说具有很大的灵活性。

2.3 基于 socket 的通信机制^[9]

在 TCP/IP 的参考模型中，应用程序处在应用层，应用层下面分别是传输层、网络层、网络接口层。传输层通过两种核心协议为应用层提供会话和数据报通信服务，即 TCP (Transmission Control Protocol, 传输控制协议) 和 UDP (User Datagram Protocol, 用户数据报协议)。TCP 协议是一种面向连接的可靠传输协

议，其工作原理是两个进程在通信前必须先建立连接；UDP 是非面向连接的，即传送数据前不需要建立连接。TCP 保证数据包的顺序性和数据正确性，UDP 不保证这一点，当然所耗费的系统资源要小于 TCP。

Socket 是一种应用程序访问通信协议的操作系统调用，它的出现使程序员可以很方便地访问 TCP/IP, 从而开发各种基于互联网的应用程序。在网络通信中 socket 可以理解为客户端或服务端的一个特殊对象，socket 通常称为套接字，用于描述 IP 地址和端口，实现客户端和服务端一对一的通信。

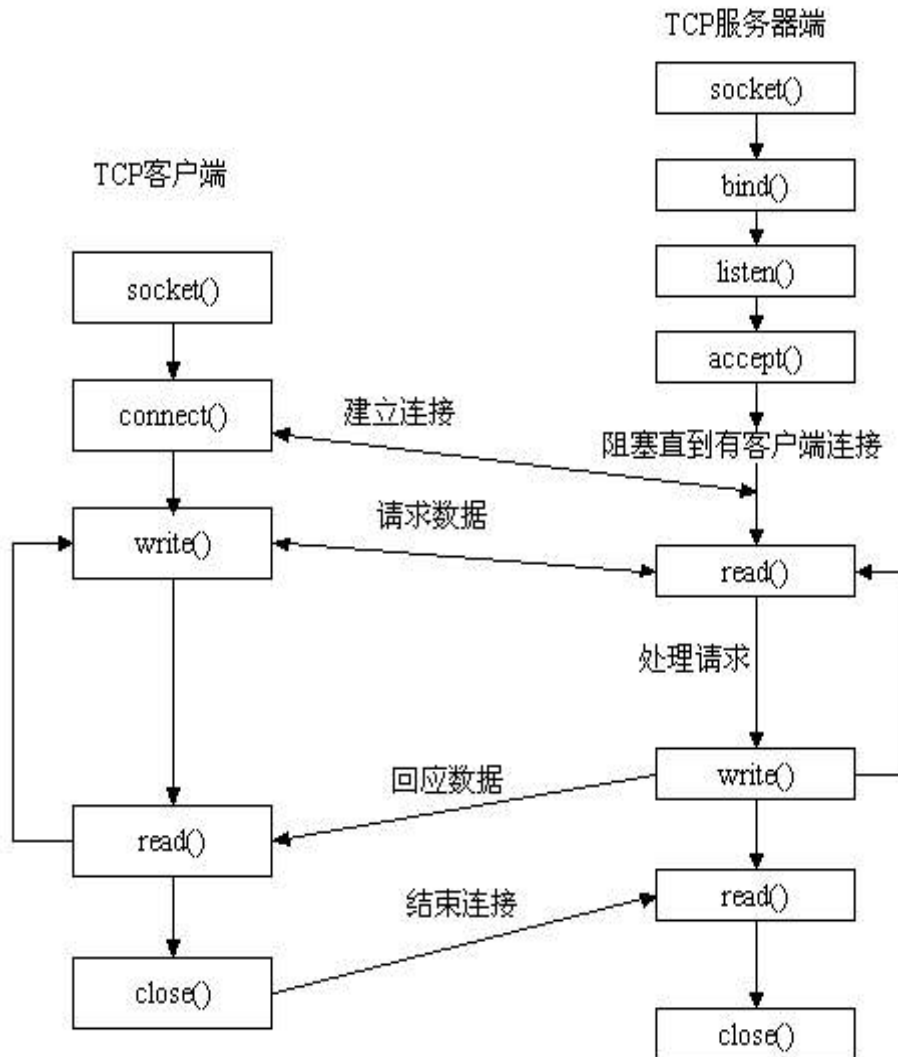


图 2-14 socket 通信交互图

服务器端先初始化 Socket，然后与端口绑定(bind)，对端口进行监听(listen)，调用 accept 阻塞，等待客户端连接。在这时如果有个客户端初始化一个 Socket，然后连接服务器(connect)，如果连接成功，这时客户端与服务器的连接就建立了。客户端发送数据请求，服务器端接收请求并处理请求，然后把回应数据发送给客户端，客户端读取数据，最后关闭连接，一次交互结束。

2. 4POSIX 规范的 MySQL 开发的 C 接口

2. 4. 1 数据结构

(1)MYSQL

连接数据库前，必须先创建 MYSQL 变量，它包含了一些连接信息等数据。

MYSQL 数据结构定义如下：

```
typedef struct st_mysql {
    NET          net;      /* Communication parameters */
    gptr         connector_fd; /* ConnectorFd for SSL */
    char
    *host,*user,*passwd,*unix_socket,*server_version,*host_info,*info,*db;
    unsigned int  port,client_flag,server_capabilities;
    unsigned int  protocol_version;
    unsigned int  field_count;
    unsigned int  server_status;
    unsigned long thread_id;      /* Id for connection in server */
    my_ulonglong  affected_rows;
    my_ulonglong  insert_id;      /* id if insert on table with NEXTNR */
    /*
    my_ulonglong  extra_info;      /* Used by mysqlshow */
    unsigned long  packet_length;
    enum mysql_status status;
    MYSQL_FIELD   *fields;
    MEM_ROOT      field_alloc;
    my_bool       free_me;      /* If free in mysql_close */
    my_bool       reconnect;    /* set to 1 if automatic reconnect */
    struct st_mysql_options options;
    char          scramble_buff[9];
    struct charset_info_st *charset;
    unsigned int  server_language;
} MYSQL;
```

(2)MYSQL_RES

MYSQL_RES 结构体中包含了查询结果集，也就是从数据库中查询到的数据。

可以使用 mysql_store_result 或 mysql_use_result 函数获得。

MYSQL_RES 数据结构定义如下：

```
typedef struct st_mysql_res {
    my_ulonglong row_count;
    unsigned int  field_count, current_field;
    MYSQL_FIELD   *fields;
    MYSQL_DATA     *data;
    MYSQL_ROWS     *data_cursor;
    MEM_ROOT      field_alloc;
    MYSQL_ROW      row;      /* If unbuffered read */
    MYSQL_ROW      current_row; /* buffer to current row */
    unsigned long  *lengths;  /* column lengths of current row */
    MYSQL          *handle;    /* for unbuffered reads */
    my_bool        eof;      /* Used my mysql_fetch_row */
} MYSQL_RES;
```

(3)MYSQL_ROW

它实际上是 char **类型，指向一个字符串数组。可以通过 mysql_fetch_row

函数获得。

定义如下：

```
typedef char ** MYSQL_ROW;
```

2.4.2 相关操作函数

(1) `MYSQL *mysql_init(MYSQL *)`

分配或初始化与 `mysql_real_connect()` 相适应的 MYSQL 对象。

(2) `MYSQL *mysql_real_connect (MYSQL *, const char *, const char *, const char *, const char *, unsigned int , const char *, unsigned long)`

`mysql_real_connect()` 尝试与运行在主机上的 MySQL 数据库引擎建立连接。在你能够执行需要有效 MySQL 连接句柄结构的任何其他 API 函数之前，`mysql_real_connect()` 必须成功完成。

(3) `int mysql_query(MYSQL *, const char *)`

执行由“Null”终结的字符串查询指向的 SQL 查询，字符串必须包含 1 条 SQL 语句，

(4) `MYSQL_ROW mysql_fetch_row(MYSQL_RES *)`

检索结果集的下一行。

(5) `void mysql_free_result(MYSQL_RES *)`

释放由 `mysql_store_result()`、`mysql_use_result()` 或 `mysql_db_query()` 等为结果集分配的内存。

(6) `void mysql_close (MYSQL *)`

关闭前面打开的连接。如果句柄是由 `mysql_init()` 或 `mysql_connect()` 自动分配的，`mysql_close()` 还将解除由 `mysql` 指向的连接句柄。

第三章 需求分析

我们要开发一个移动终端上的流媒体播放客户端软件，播放器要具有选择视频的功能，这样我们能决定选择播放不同视频。

为了保证有一个稳定的视频流数据源，我们要搭建一个流视频服务器，在上面放置若干视频文件，作为视频数据源。视频文件的元信息，如 URL、视频名称等数据储存在数据库里。

为了能够对测量过程进行有效地检测和记录，我们对用户进行管理，要求用户进行注册，只有合法用户才能获得系统响应并正常播放视频。其中用户的数据储存在服务器端数据库里。

3.1 需要具有的功能

图 3-1 是功能图示，展示了用户能进行的所有操作。

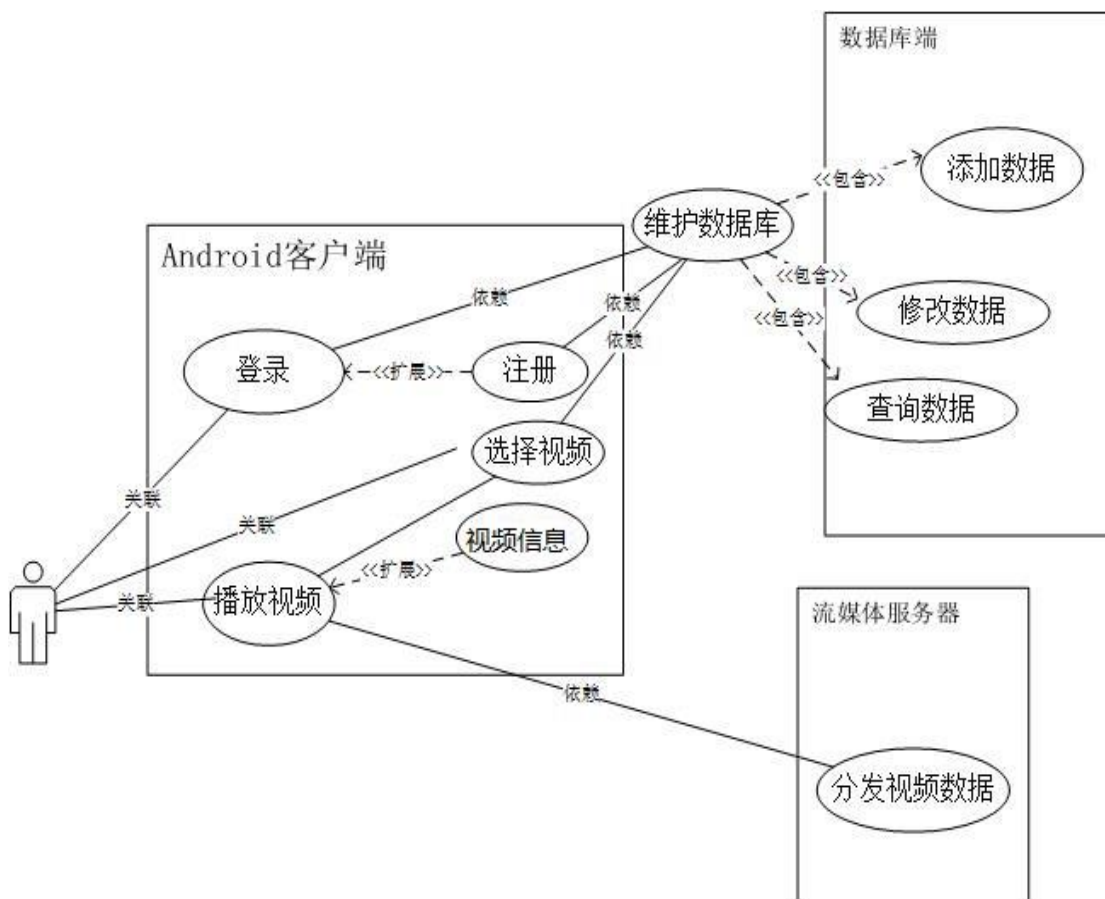


图 3-1 系统功能

从上图可以看出，用户在客户端能够执行的操作包括注册、登录、选择视频、播放视频、查询视频信息。其中播放视频功能的实现依赖流视频服务器提供视频数据源。其余四个功能均需要数据库的参与。

下面几小节依次对这几个功能进行讲述。

3.2 用户注册和登录

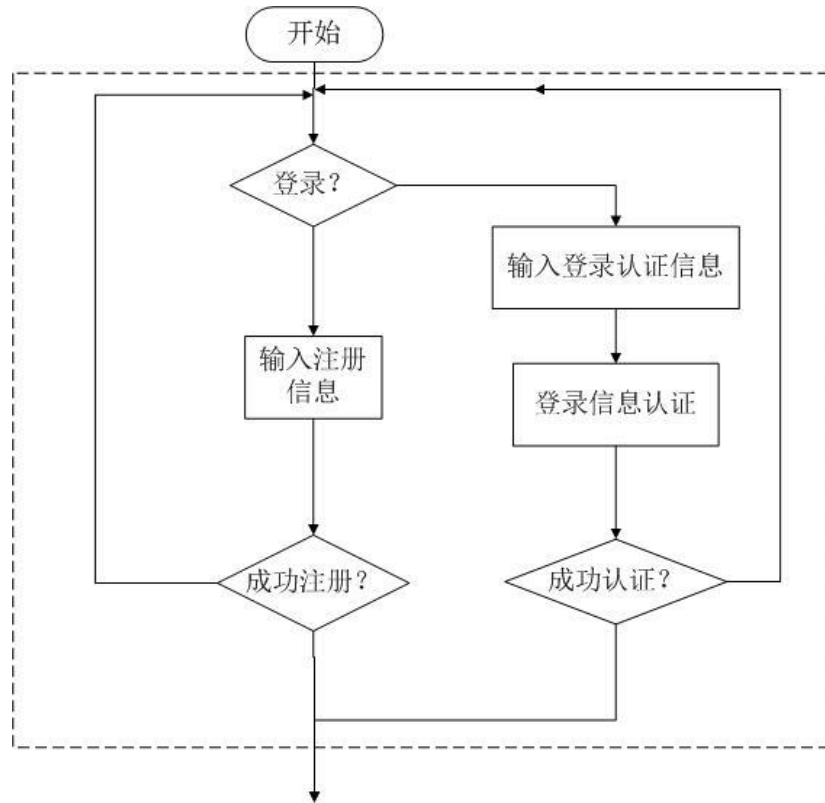


图 3-2 注册和登录流程

用户使用手机客户端需要登录，登录时要在登录界面的表单中输入正确的用户名和密码，否则不允许登录。用户如果没有账号，就需要注册一个账号。注册时除需要在注册界面的表单中输入用户名和密码外，还要填入昵称和邮箱，以便于系统对用户的控制。

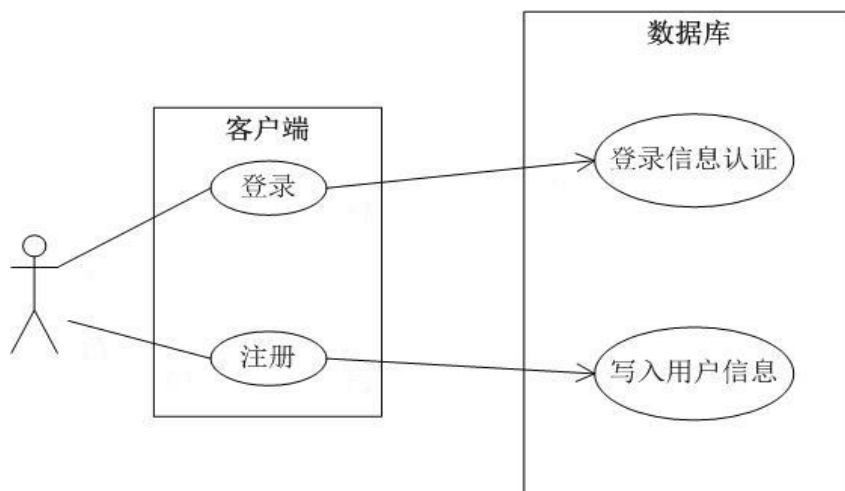


图 3-3 注册登录过程与数据库

注册和登录过程提交的信息都要传输到服务器端，其中注册的信息需要写入数据库；登录的信息需要与数据库中已有的账户信息进行对比，以确认是否是已有账户，并返回认证状态。

3.3 视频的选择

用户登录后，进入视频选择的界面，此时客户端立即向服务器端请求所有视频的名称列表并显示在客户端应用的界面上。用户可以滑动浏览所有视频，并选择自己喜欢的视频来播放。当用户选择某视频时，客户端将向服务器端请求该视频的 URL 信息，这是视频资源的网络地址。过程如图 3-4 所示：

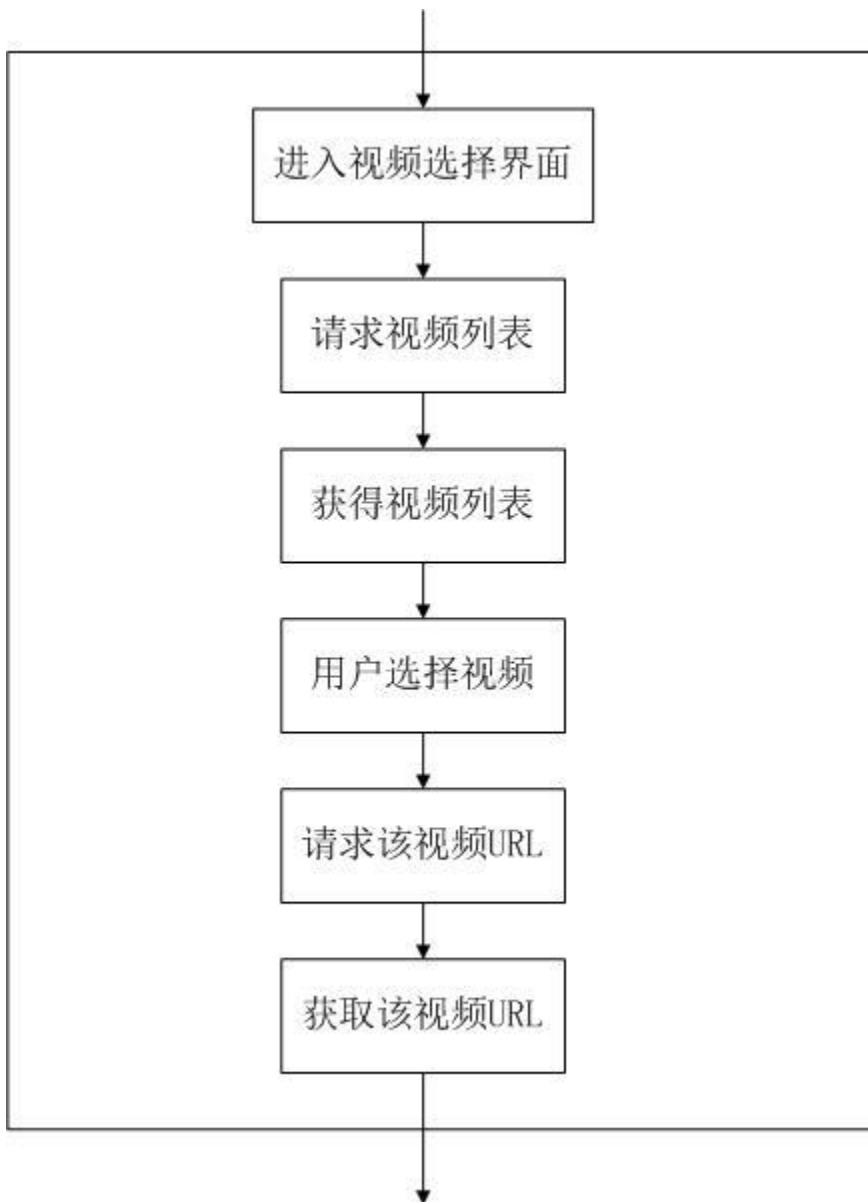


图 3-4 视频选择流程

3.4 基于 Android 的流媒体播放器

播放器在 Android 手机上运行，不但能够正常播放常规的网络视频，还能过控制其停止、继续播放、前进、后退。

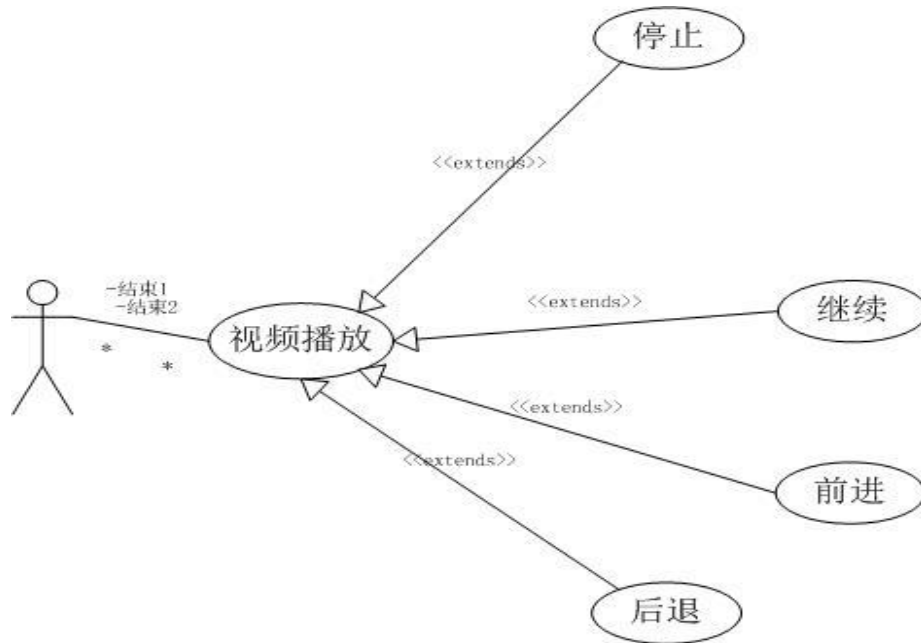


图 3-5 播放器操作

3.5 查看当前视频信息

如果用户在观看网络视频时对视频本身很感兴趣，可以请求视频的详细信息，如视频介绍等，这些信息是预先存储在服务器端数据库中的。

视频信息的显示应当不影响视频播放进程。

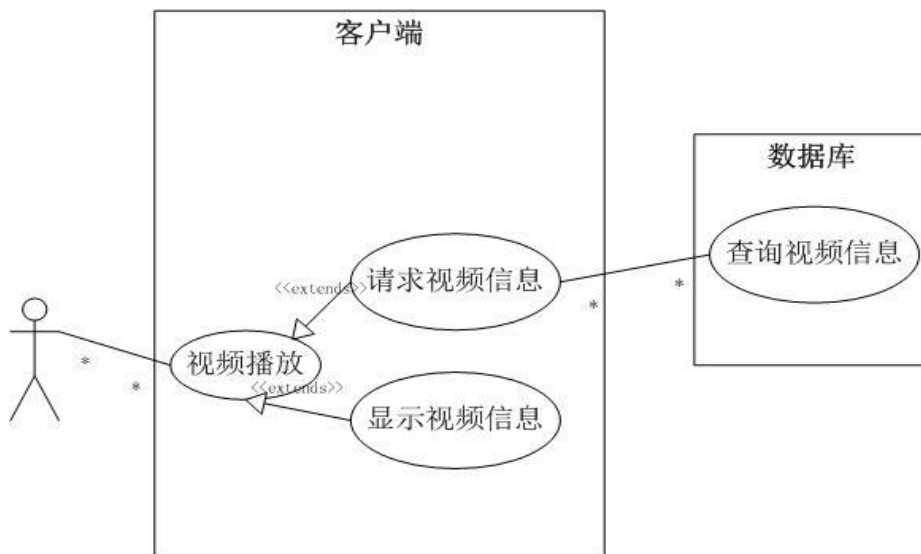


图 3-6 查询视频信息

第四章 开发环境与平台搭建

4.1 开发及运行环境

4.1.1 硬件平台

表 4- 1 硬件参数

硬件	技术参数
移动设备	S5PV210 处理器, 1GHz, RAM 512M, 800*480 液晶屏, Android 2.3.1
服务器	Intel(R) Core(TM) i5 CPU,主频 3.2GHz, RAM 4GB,Ubuntu 12.04

4.1.2 开发工具

表 4- 2 软件参数

软件开发平台	参数	功能
Eclipse	eclipse-java-galileo-linux-gtk	Java 开发平台
Android ADT	adt-bundle-linux-x86	Android 开发工具
NDK	android-ndk-r9d-linux-x86	Android 本地开发工具
GCC	gcc-4.5.1	C 语言编译工具
MySQL	MySQL for Linux 5.6.4	数据库

4.2 Android 系统开发的平台搭建

Android 系统内核采用的是 Linux 内核, 因此在 Linux 环境下搭建开发平台更容易操作, 当然在 Windows 下进行开发也是可以的。下面介绍一下 Android 开发所需要安装的开发包, 主要包括 Eclipse, JDK (Java Development Kit), Android SDK(Software Development Kit), ADT(Android Development Tools) 和 Android NDK(Native Development Kit)³。

①Eclipse: Eclipse 也是一个开源的、基于插件的、面向多种语言的开发平台, 目前主要用于 Java 语言开发, 是 java 开发人员常用的 IDE(Integrated Development Environment, 集成开发环境)。Eclipse 的最新版本是 Luna, 可以在 Eclipse 官网上进行免费下载, 解压后就可以使用, 可以通过 Eclipse 的 Help 工具中的 Install New Software 安装 Android 的一些开发工具, 如 Android NDK, ADT, 还可以指定 Android SDK 的目录。

②JDK: JDK 主要是用来进行 java 开发需要安装的开发包, 是 java 是整个

³ <http://blog.csdn.net/golden1314521/article/details/23018913>

Java 的核心，包括了 Java 运行环境（Java Runtime Environment），一堆 Java 工具和 Java 基础的类库（rt.jar）。不论什么 Java 应用服务器实质都是内置了某个版本的 JDK。JDK 使用前需要设置环境变量。

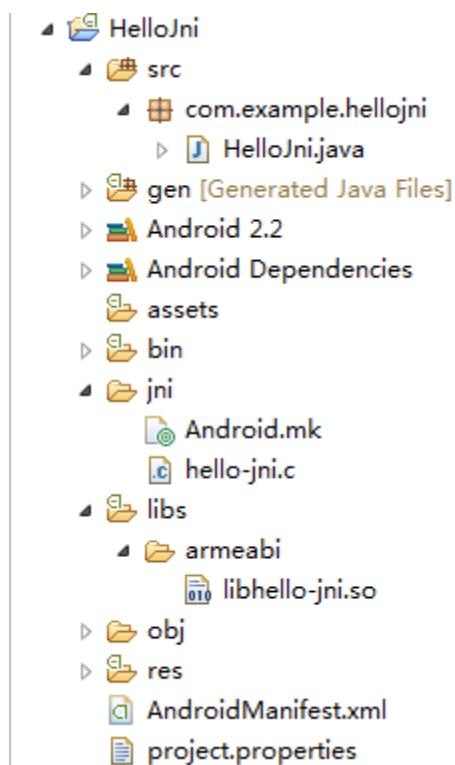


图 4-1 Android NDK 开发工程图

③Android SDK; Android SDK 是 Google 公司推出的开发 Android 软件的工具包，它包括 Android 开发相关的各种工具软件和 jar 包。Android SDK 可以直接从官网上下载并解压，然后在 Eclipse 中的 Android Preferences SDK Location 中选择 Android SDK 解压的文件的路径。

④Android NDK: NDK 提供了一系列的工具，帮助开发者快速开发 C（或 C++）的动态库，并能自动将 so 和 java 文件一起打包成 apk。这些工具极大地简化了开发者的工作，使其专注于高层次的设计，而不用总是在处理重复的、大量的、没有创意的细致末节的。NDK 集成了交叉编译器，并提供了相应的 mk 配置文件来屏蔽平台、CPU、API 等差异，开发人员只需要简单修改 mk 文件（指出“编译的文件源”、“编译参数选取”等），就可以创建出 so 库文件。NDK 提供了一份稳定、功能有限的 API 头文件声明。通过 NDK 和 JNI 中间层，Android 程序就可以直接调试、调用 C/C++代码。

⑤ADT: Android SDK 和 NDK 不能再 Eclipse 中直接使用，必须借助于 ADT 提供的调用接口。ADT 可以在线安装，也可以下载后离线安装。

搭建好开发环境后，就可以进行 Android 开发了。图 4-1 是 Android NDK 开发的一个工程示例的内部结构，可以看到 .java 文件、.c 文件、Android 配置文件、相关的库。

其中 HelloJni.java、Android.mk、hello-jni.c，libhello-jni.so 之间的关系可以用图 4-2 表示：

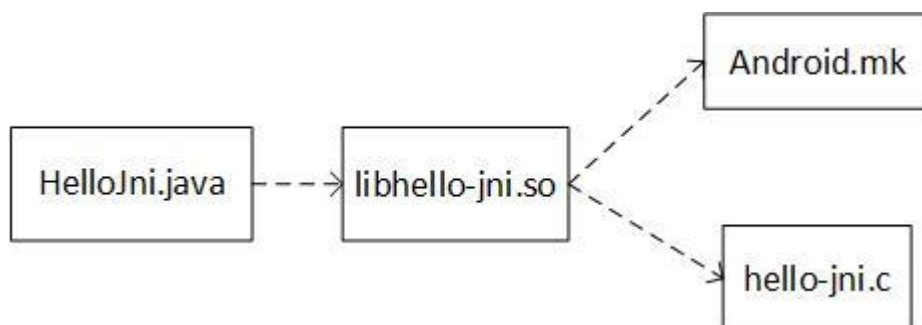


图 4-2 JNI 工程文件依赖关系图

上图表示，HelloJni.java 文件依赖于可执行库文件 libhello-jni.so，表现为 HelloJni.java 调用了 libhello-jni.so；而 libhello-jni.so 依赖于 Android.mk 和 hello-jni.c 文件，因为前者是由后者编译生成的。

第五章 系统概要设计

5.1 系统总体设计

根据需求，要实现基于移动互联网的流媒体点播系统，将系统划分为四大部分，依次是移动终端、流媒体服务器、Socket 服务器和数据库服务器。如图 5-1 所示：

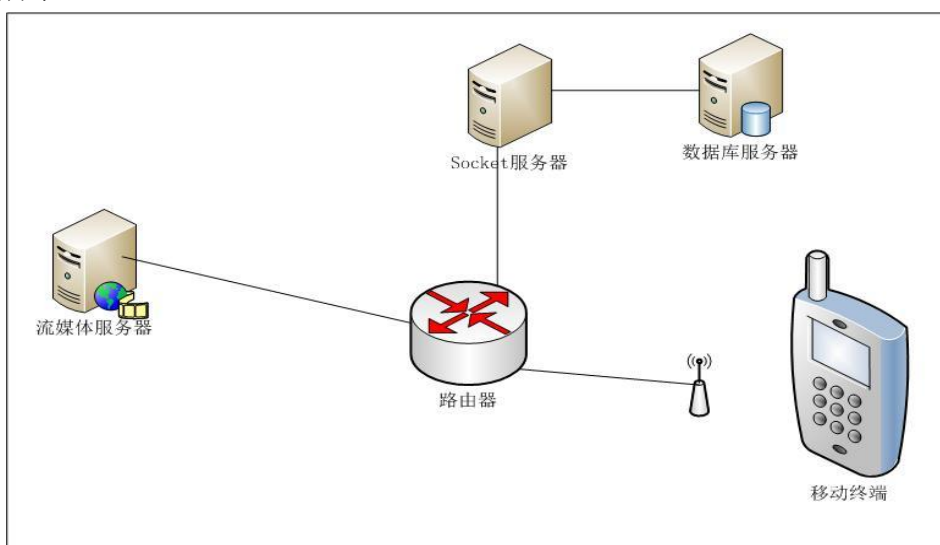


图 5-1 流视频点播系统架构

从上图可以看出，移动终端通过无线网络（如 WiFi、2G、3G）连接路由器，继而直接与流媒体服务器和 socket 服务器相连接，这里所说的直接是相对于数据库服务器来说的，可以看到移动终端不直接与数据库服务器相连，而是通过 socket 服务器相连接。可以看出 socket 服务器处于很重要的地位，它承担了移动终端与数据库服务器之间请求和应答的中转任务，如有必要，它还会对数据进行处理。

5.2 数据库服务器的设计

从第三章的内容可知，客户端具有执行注册、登录认证的功能，所以数据库要能够存储用户信息，所以我们需要用户信息数据表；客户端能够查询视频信息，所以我们需要视频信息数据表用来存储视频信息。

用户信息数据表和视频信息数据表初步设计如下：

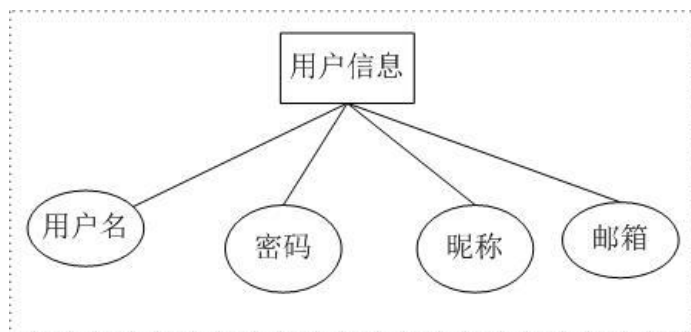


图 5-2 用户信息 E-R 图

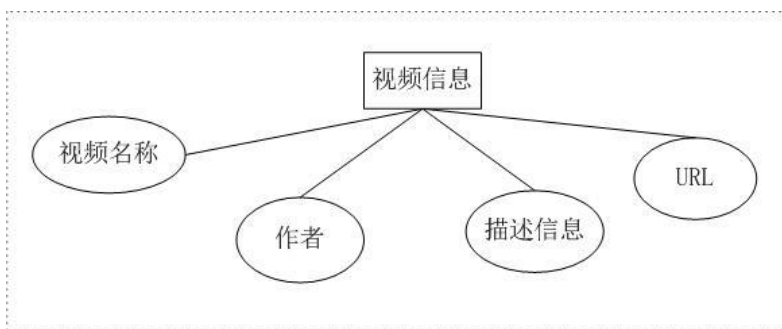


图 5-3 视频信息 E-R 图

5.3 流媒体服务器的设计

从第二章关于三大流媒体服务器平台的探讨中我们发现，作为流媒体服务器，Darwin Streaming Server 是完全开源的，任何开发人员都可以对其功能进行扩充甚至修改服务器的核心功能。因而我们选用其作为我们的流媒体服务器，这方便了我们将来对流媒体服务器的进一步研究，并在此基础上对流媒体服务器进行二次开发。

5.4 移动客户端的设计

5.4.1 模块划分

图 5-4 是移动客户端的模块划分图：

UI 是主控模块，是程序的控制核心，它与用户直接接触，是用户操作的提供者和响应者。用户通过 UI 提供的操作接口调用其他模块，如 UI 转换模块、视频播放模块、视频信息刷新模块、用户信息模块、注册模块、登录模块、视频选择模块等。其中除 UI 转换模块和视频播放模块外，其余模块都依赖于 socket 通信模块提供的服务。各个模块的作用如表 5-1 所示。

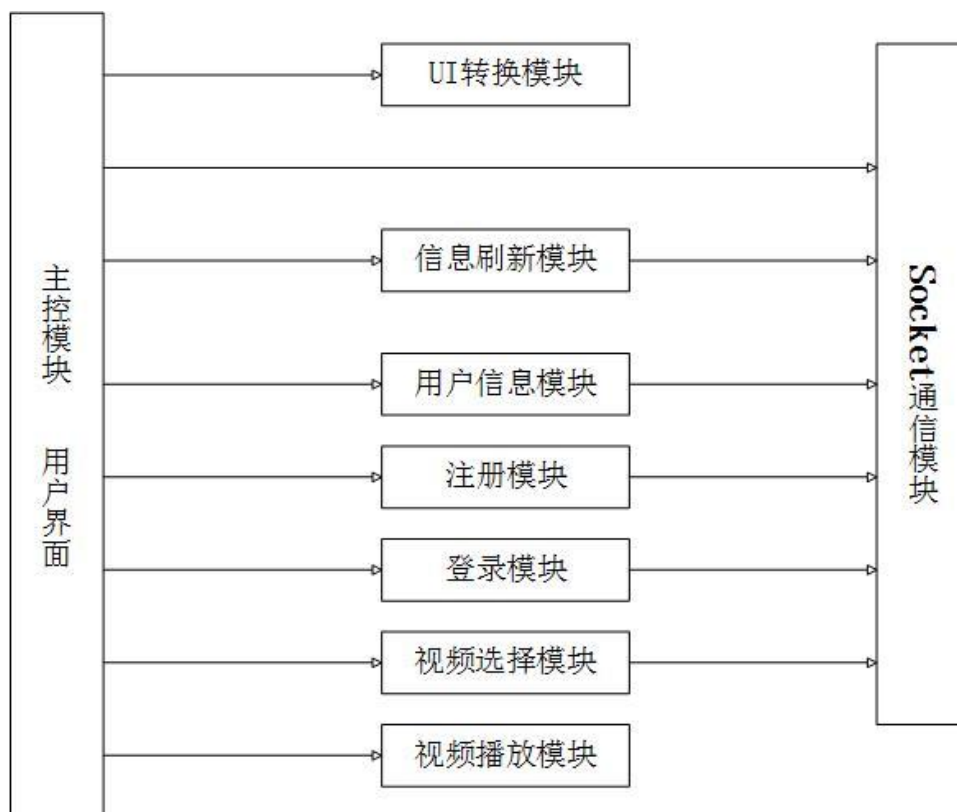


图 5-4 移动客户端功能模块设计

表 5-1 移动客户端各模块功能

模块划分	功能
用户界面	负责处理用户界面的切换
用户注册界面	接收用户注册信息，将用户信息写入数据库
用户登录界面	接收用户登录信息，并与数据库信息的进行核对
视频选择界面	接收选择的视频信息，进而获得其 URL 地址
视频播放模块	播放网络视频
视频信息刷新模块	应用户请求，定时刷新正在播放的视频的信息
用户信息模块	储存用户信息
Socket 通信模块	主要用来连接 socket 服务器，进而对数据库进行操作

5.4.2 UI 界面设计

我们设计 4 个用户界面，依次是用户登录界面、用户注册界面、视频选择界面和视频播放界面。用户首先看到的是登录界面，如果用户已有账号，则输入账号和密码即可跳转到视频选择界面。如果用户没有账号，则可选择进入注册界面去申请一个用户账号，进而进入视频选择界面。在视频选择界面，用户可以选择播放自己喜欢的视频，播放过程则在视频播放界面。

UI 界面转换关系如图 5-5 所示：

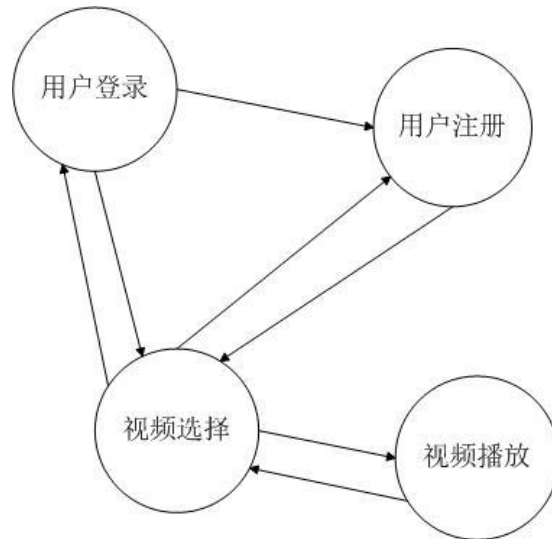


图 5-5 UI 转换图

4 个 UI 用到了不同的功能模块，使用情况如表 5-2 所示。

表 5- 2 模块使用图

用户界面	功能模块
用户登录	用户登录模块、UI 转换模块、Socket 通信模块、用户信息模块
用户注册	用户注册模块、UI 转换模块、Socket 通信模块、用户信息模块
视频选择	视频选择模块、UI 转换模块、Socket 通信模块、用户信息模块、
视频播放	视频播放模块、UI 转换模块、视频信息刷新模块、Socket 通信模块、用户信息模块

5.5 移动客户端和 socket 服务器的规约

移动客户端有多个功能需要用到数据库，而且需要查询数据、插入数据、更新数据等多种操作。Socket 服务器被设计用来作为移动客户端和数据库端交互的中介，移动终端向数据库发出的复杂的请求逻辑通过 socket 服务器后，转换成了简单的查询、插入、更新数据的操作。图 5-6 描述了这种联系。

可以看到手机客户端向 socket 服务器请求的操作有 5 类，分别是：写入注册信息到数据库，用户登录信息认证，向数据库请求所有视频的名称列表，请求某视频的统一定位符(URL)，请求某视频的所有信息。Socket 服务器不仅对来自客户端的请求指令进行解析，还对来自数据库的数据进行处理。

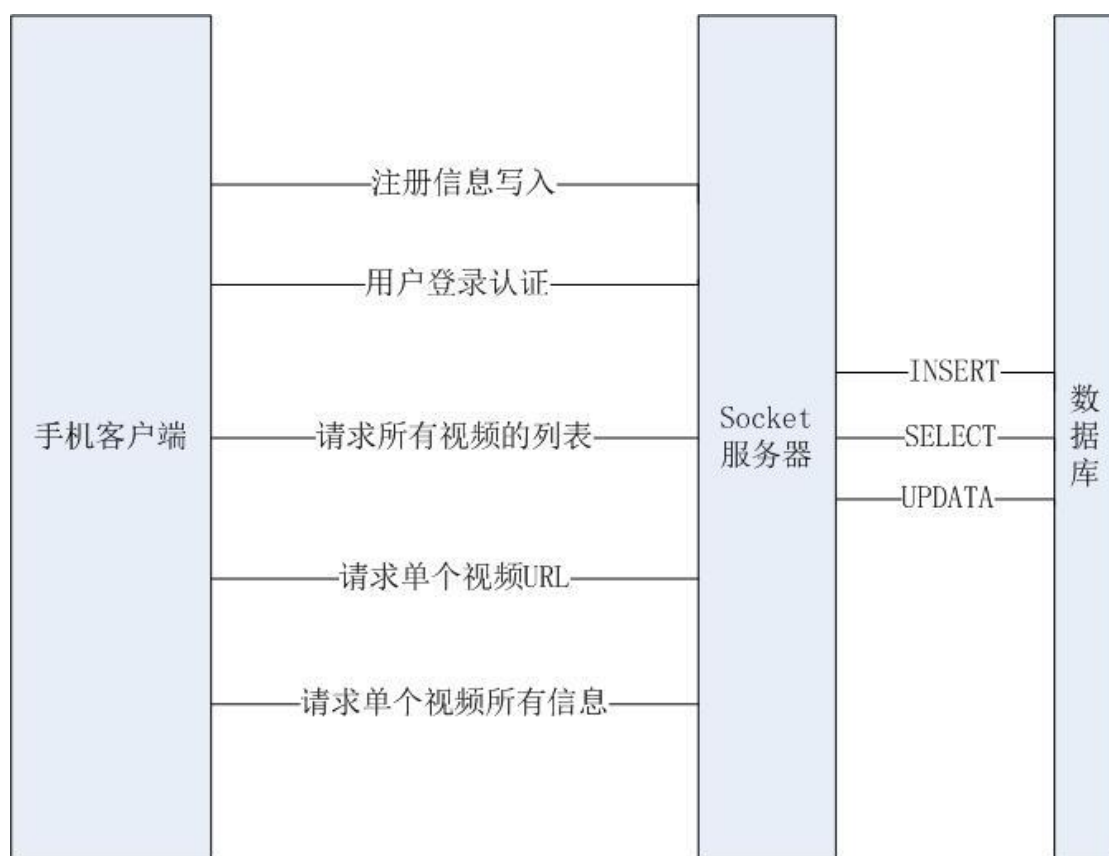


图 5-6 手机客户端和 socket 服务器的规约

第六章 系统详细设计和实现

6.1 Android 手机客户端的实现

6.1.1 Android 客户端工程类图

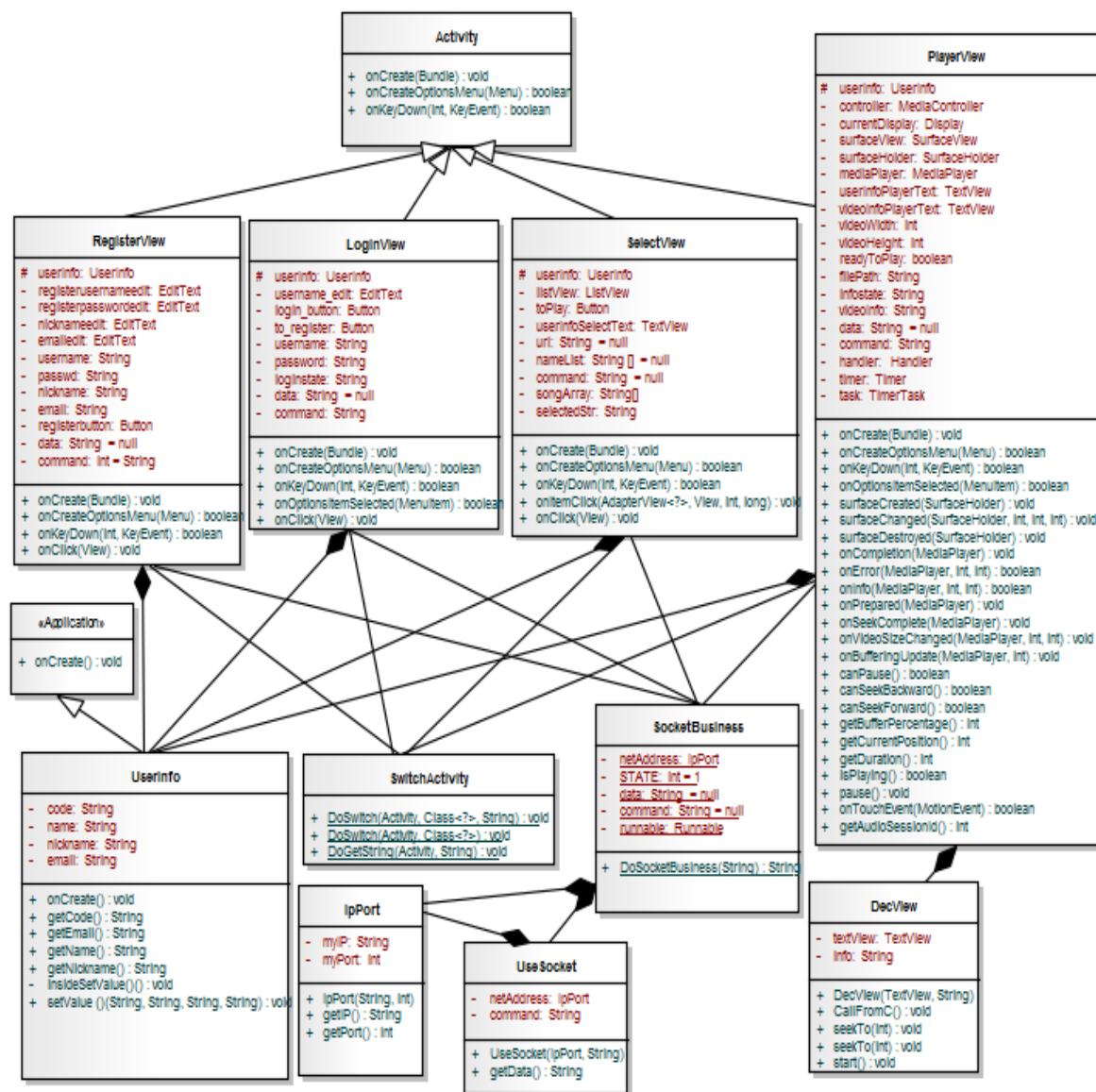


图 6-1 手机客户端工程类图

图 6-1 是 Android 手机客户端程序的类图结构，可以看到居于中心地位的是 4 个 UI 类，它们是 RegisterView 类、LoginView 类、SelectView 类和 PlayerView 类，分别表示注册界面类、登录界面类、视频选择界面类和视频播放界面类，负责同名的界面显示。这四个 UI 类与外部的关系很类似，例如它们都继承于 Activity 基类，都与另外的三个类关联，这三个类依次是 SwitchActivity

类，表示切换活动，负责不同界面之间的切换；SocketBusiness 类，表示 socket 事务，负责与 socket 服务器交互；UserInfo 类，表示用户信息，负责向 UI 类提供当前使用本软件的用户信息。

6.1.2 各个类内部实现的解析

从图 6-1 看类，往往感觉各类之间关系错综复杂，无法得知类的内部机制，因此不容易理解类之间存在关系的实现机制。本小节将对重要的类进行进一步讲述。

①IpPort 类。IpPort 类代表了一个 socket 逻辑地址，包括 IP 地址和开放的端口，其内部成员如表 6-1 所示：

表 6-1 IpPort 类成员

成员	类型	含义
myIP	字符串	IP 地址
myPort	整型	端口
IpPort(String,int)		构造函数，用于初始化对象
getIP ()	字符串	用于获取 IP 地址
getPort ()	整型	用于获取端口

②UseSocket 类。UseSocket 类封装了处理 socket 事务中与服务器端交互的逻辑控制，内部成员列表如表 6-2 所示。

表 6-2 UseSocket 类成员

成员	类型	含义
netAddress	IpPort	UseSocket 内部包含的一个 IpPort 对象
command	String	Socket 事务的操作指令
UseSocket(IpPort,String)		构造函数，用于初始化对象
getData()	String	从 socket 服务器获取数据，内部封装了处理 socket 交互的事务处理的逻辑

③SocketBusiness 类。SocketBusiness 类对使用了 IpPort 类和 UseSocket 类，对它们进行封装，目的是使其能够在 Android 中调用。

表 6-3 SocketBusiness 类成员

成员	类型	含义
netAddress	IpPort	内部包含的一个 IpPort 对象
STATE	Int	Socket 事务处理状态，值取 1 时表示未处理完成，取 0 时表示处理完成
data	String	Socket 事务处理完成后返回的数据包

command	String	Socket 事务的操作指令
runnable	Runnable	开启子线程，在子线程中启动 socket 事务。由于 socket 事务需要连接网络，而 Android 禁止主线程执行网络操作，所以只能在子线程中处理 socket 事务。
DoSocketBusiness(String)	String	启动 socket 事务处理逻辑

为了进一步说明 SocketBusiness、IpPort 类和 UseSocket 类之间的调用关系，图 6-2 展示了连接 socket 服务器时三个类内部的动态：

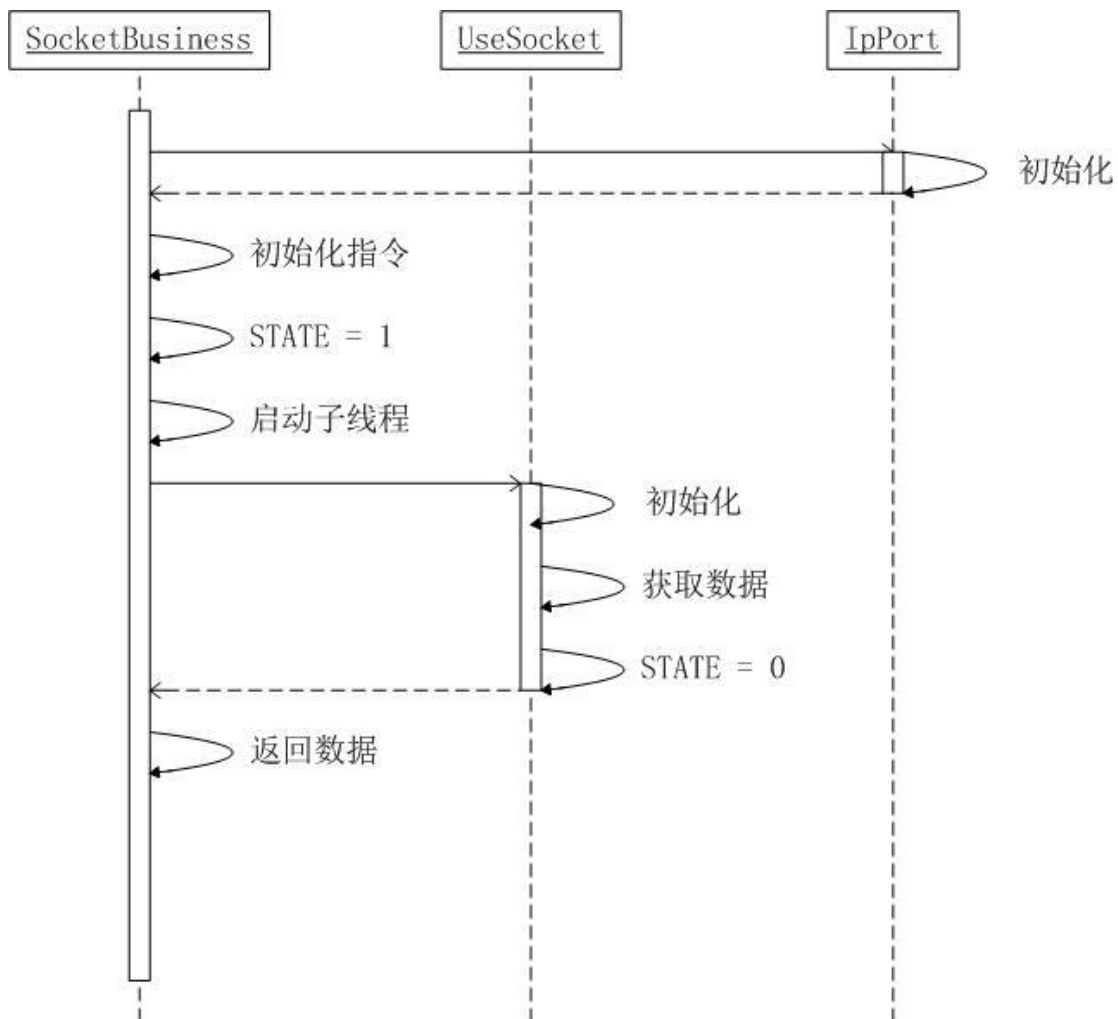


图 6-2 socket 服务相关的三个类

④UserInfo 类。UserInfo 类封装了用户信息，从图 6-1 中可以看到 UserInfo 类继承了 Application 类，而 Application 类能够实现数据传递、数据共享、数据缓存等功能，所以 UserInfo 对象在能够实现在整个程序运行过程中共享用户信息，以实现对用户的追踪。

表 6- 4 UserInfo 类成员

成员	类型	含义
name	String	用户名
code	String	用户密码
nickname	String	用户昵称
email	String	用户电子邮件
onCreate ()	void	覆盖的父类方法，用于初始化类对象
getName ()	String	返回用户名
getCode ()	String	返回用户密码
getNickname ()	String	返回用户昵称
getEmail ()	String	返回用户电子邮件
setValue (String, String, String, String)	void	设置成员变量的值

⑤SwitchActivity 类。SwitchActivity 类负责 UI 之间的切换，需要说明的是每个 UI 类都继承于 Android 中的 Activity 类。本类没有成员变量，所有的方法都是静态方法，可以不经初始化对象即可使用。其中 DoSwitch 方法发生了参数重载，负责执行 UI 切换。

表 6- 5 SwitchActivity 类成员

成员	类型	含义
DoSwitch (Activity, Class<?>)	void	负责 UI 切换，第一个参数是当前 UI 类，第二个参数是目的 UI 类
DoSwitch (Activity, Class<?>, String)	void	重载方法，负责 UI 切换，第三个参数是当前 UI 类传送给目的 UI 类的数据
DoGetString (Activity, String)	String	在目的 UI 类中使用，负责取出从原 UI 传送来的数据

⑥DecView 类。DecView 类负责文本的动态显示，它的构造函数有两个参数，第一个参数是一个 TextView，这是 UI 界面上显示文本的控件，第二个参数是 String，这是要在 TextView 控件上显示的字符串。

表 6- 6 DecView 类成员

成员	类型	含义
textView	TextView	Android 界面的一种控件，可以显示文本，即在此控件上显示字符串，并可以设置显示字符串的颜色、字体等
info	String	要显示的字符串

DecView(Textview, String)		构造函数
Decorate()	native void	本地方法，其具体实现在本地代码中，即 C 代码
CallFromC()	void	这是个 Java 方法，但是要被 C 代码调用

DecView 类实现了对字符串颜色的动态刷新，其实现机制如图 6-3 所示：

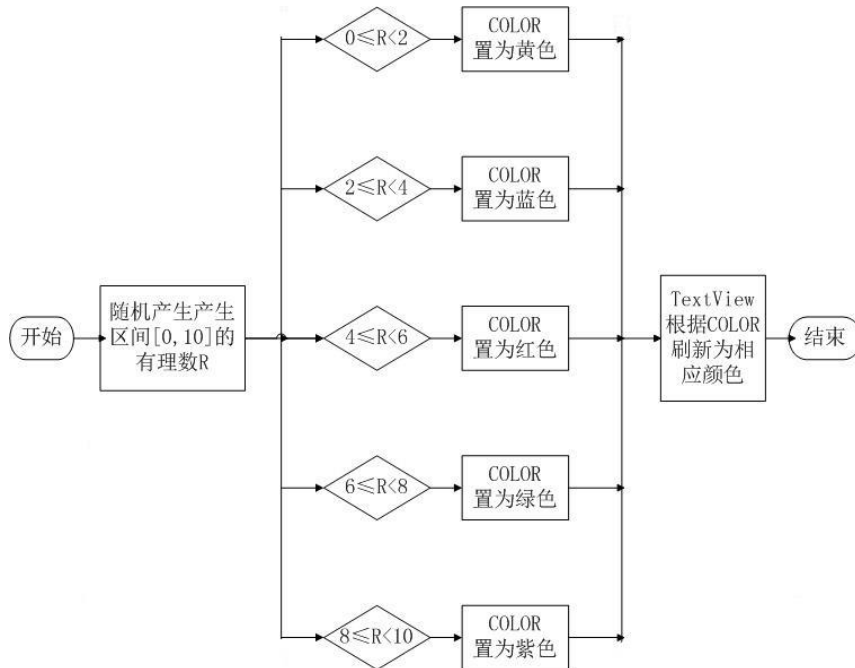


图 6-3 动态刷新颜色机制

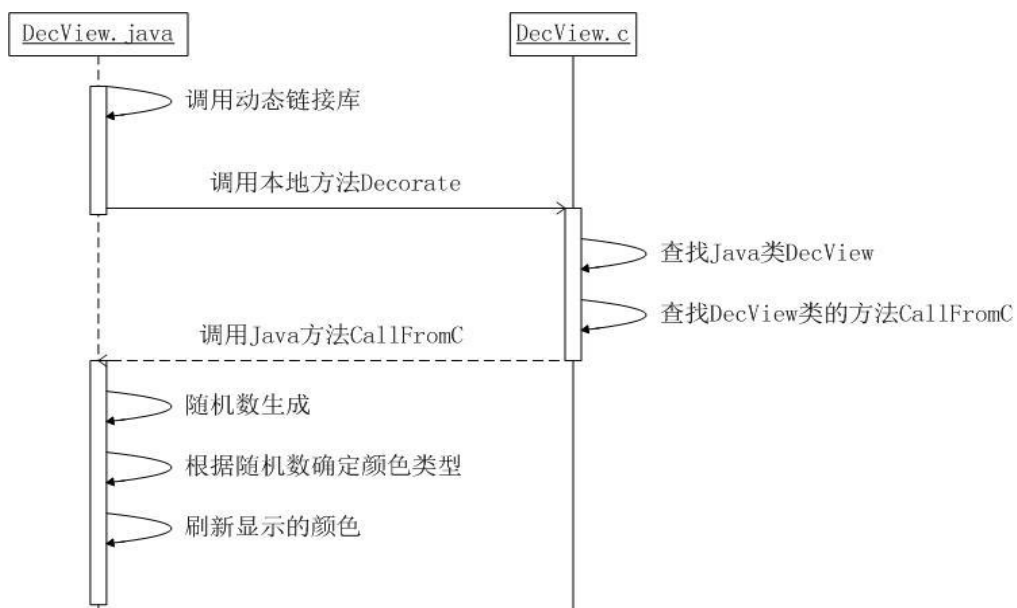


图 6-4 DecView 类的实现

DecView 类的实现用到了 JNI 方法，用到了 NDK 编程，简单地说就是用 C 语

言写底层实现，上层的 Java 类通过 JNI 提供的接口调用 C 源码。这里 Java 类文件名是 DecView.java，C 源码文件是 DevView.c。DecView.java 和 DevView.c 之间的调用序列图如图 6-4 所示。

⑦RegisterView 类。RegisterView 类是负责用户注册界面的类，它的内部成员变量和成员函数比较复杂，我们展示其类图关系如图 6-5 所示。

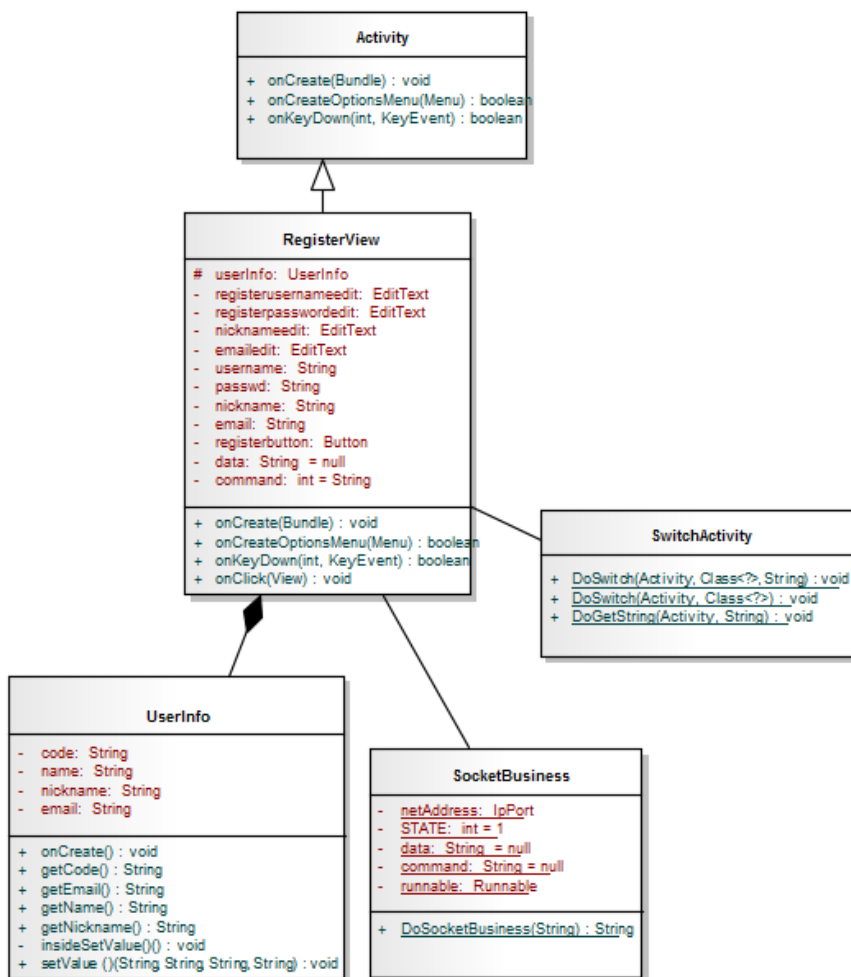


图 6-5 RegisterView 类图关系

可以看到 RegisterView 类继承了 Android 中的 Activity 类，这是 Android 中与界面控制直接相关的类；RegisterView 类还与 UserInfo 类、SocketBusiness 类和 SwitchActivity 类相关量，以分别实现其跟踪用户、连接 socket 服务器和切换到其它用户界面的功能。它利用 socket 服务器将用户注册信息写入数据库。

⑧LoginView 类。LoginView 类是负责用户注册界面的类，与 Register 类类似，它的内部成员变量和成员函数比较复杂，我们展示其类图关系如图 6-6 所示。

可以看到 LoginView 类继承了 Android 中的 Activity 类，这是 Android 中与界面控制直接相关的类；LoginView 类还与 UserInfo 类、SocketBusiness 类和 SwitchActivity 类相关量，以分别实现其跟踪用户、连接 socket 服务器和切换

到其它用户界面的功能。它利用 socket 服务器将用户登录信息，包括用户名和密码与数据库保存的用户信息对比，以确定登录用户信息是否合法。

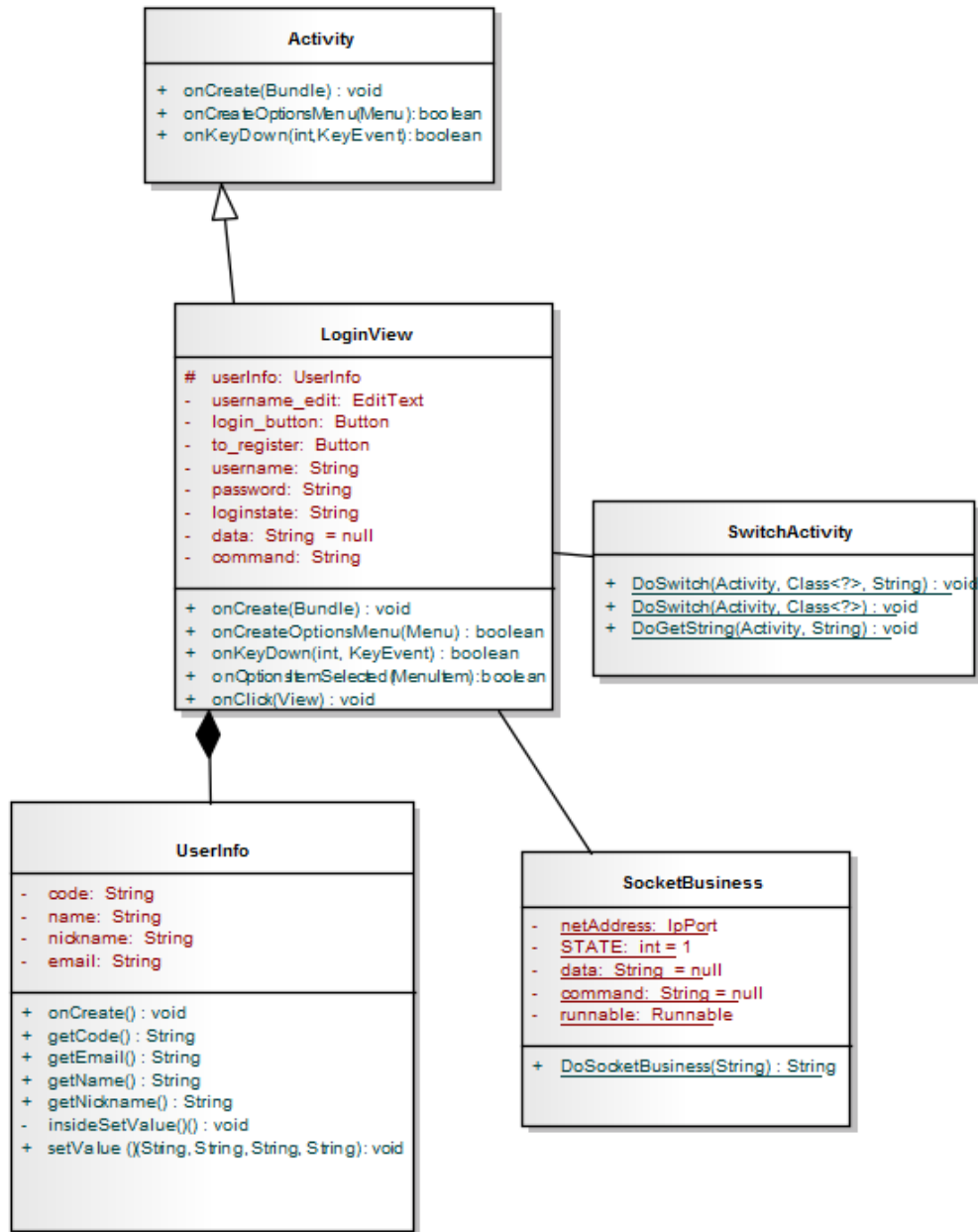


图 6-6 LoginView 类图关系

⑨SelectView 类。SelectView 类是负责视频选择界面的类，与 RegisterView 类和 LoginView 类似，它的内部成员变量和成员函数比较复杂，我们展示其类图关系如图 6-7 所示。

可以看到 SelectView 类继承了 Android 中的 Activity 类，这是 Android 中与界面控制直接相关的类；LoginView 类还与 UserInfo 类、SocketBusiness 类和 SwitchActivity 类关联，以分别实现其跟踪用户、连接 socket 服务器和切换到其它用户界面的功能。它先利用 socket 服务器来获取所有的视频名称的列表，并将此列表的信息显示在当前界面上来供用户选择；当用户选择某视频后，它利

用 socket 服务器来获取选定视频的 URL 信息,以便于进入播放界面时根据此 URL 信息链接视频服务器获取对应的视频。

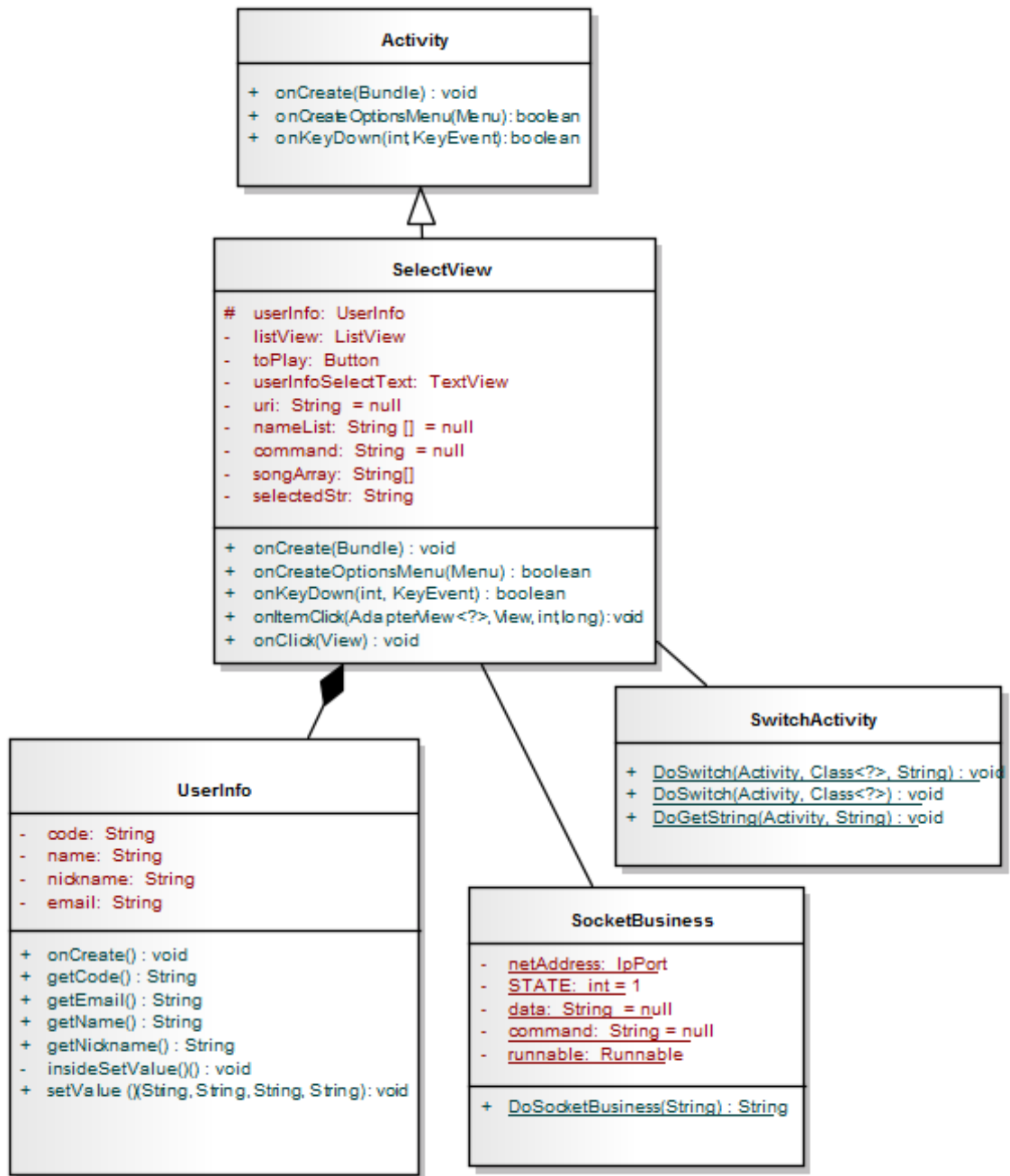


图 6-7 SelectView 类图关系

⑩PlayerView 类。PlayerView 类尤其复杂,可以看到它拥有更多的成员变量和成员函数。它与 UserInfo 类、SocketBusiness 类和 SwitchActivity 类关联,因而具有跟踪用户信息、切换到其它界面和与 socket 服务器连接的功能。它利用 socket 服务器来获取某个视频的所有信息,如视频名称、作者、视频简介和 URL 等信息。它与 DevView 类相关联,它利用这个类来完成动态刷新视频信息的颜色的工作。

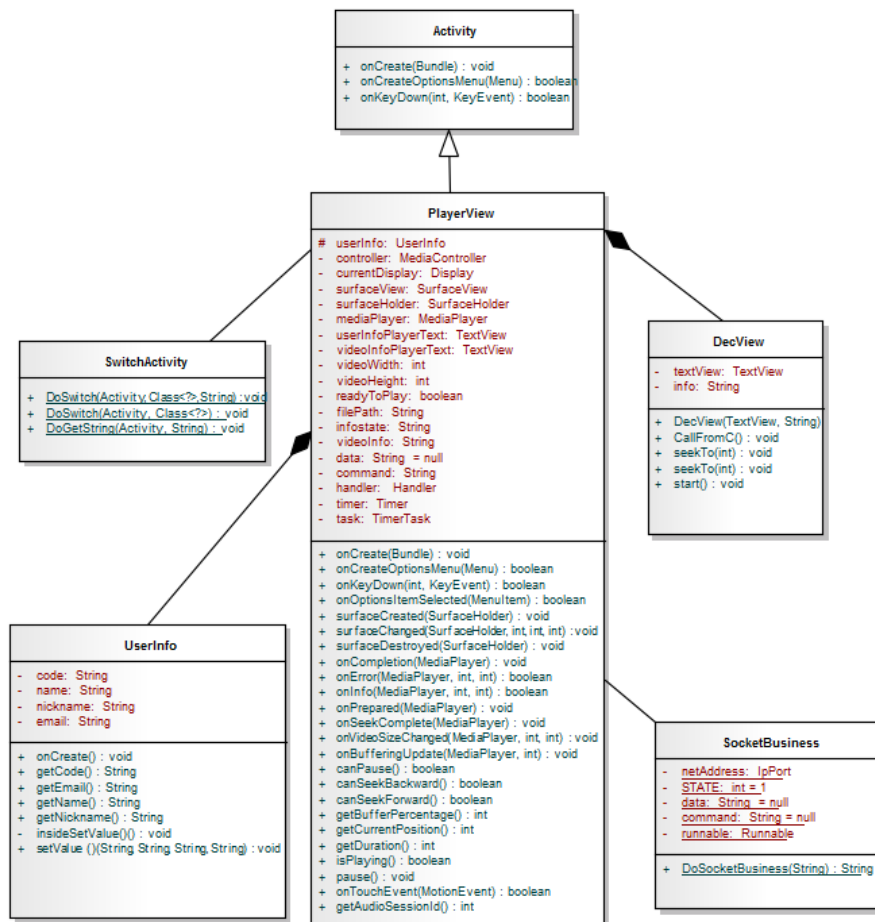


图 6-8 PlayerView 类图关系

6.2 Socket 服务器端的开发

Socket 服务器采用 C 语言开发，它是运行在 Linux 系统上的一个守候进程，一直监听一个固定的端口。当监听到来自客户端的请求数据包时，程序启动一个子线程，把处理此请求的工作交给这个子线程，然后继续监听此端口。子线程被创建后，首先解析请求数据包，确定要执行的操作，然后进入响应的处理模块完成应答数据的获取，然后对获取到的数据进行封装，将应答数据封装成应答数据包，最后将数据包发送给客户端，结束子线程。

Socket 服务器工作流程如图 6-9 所示。

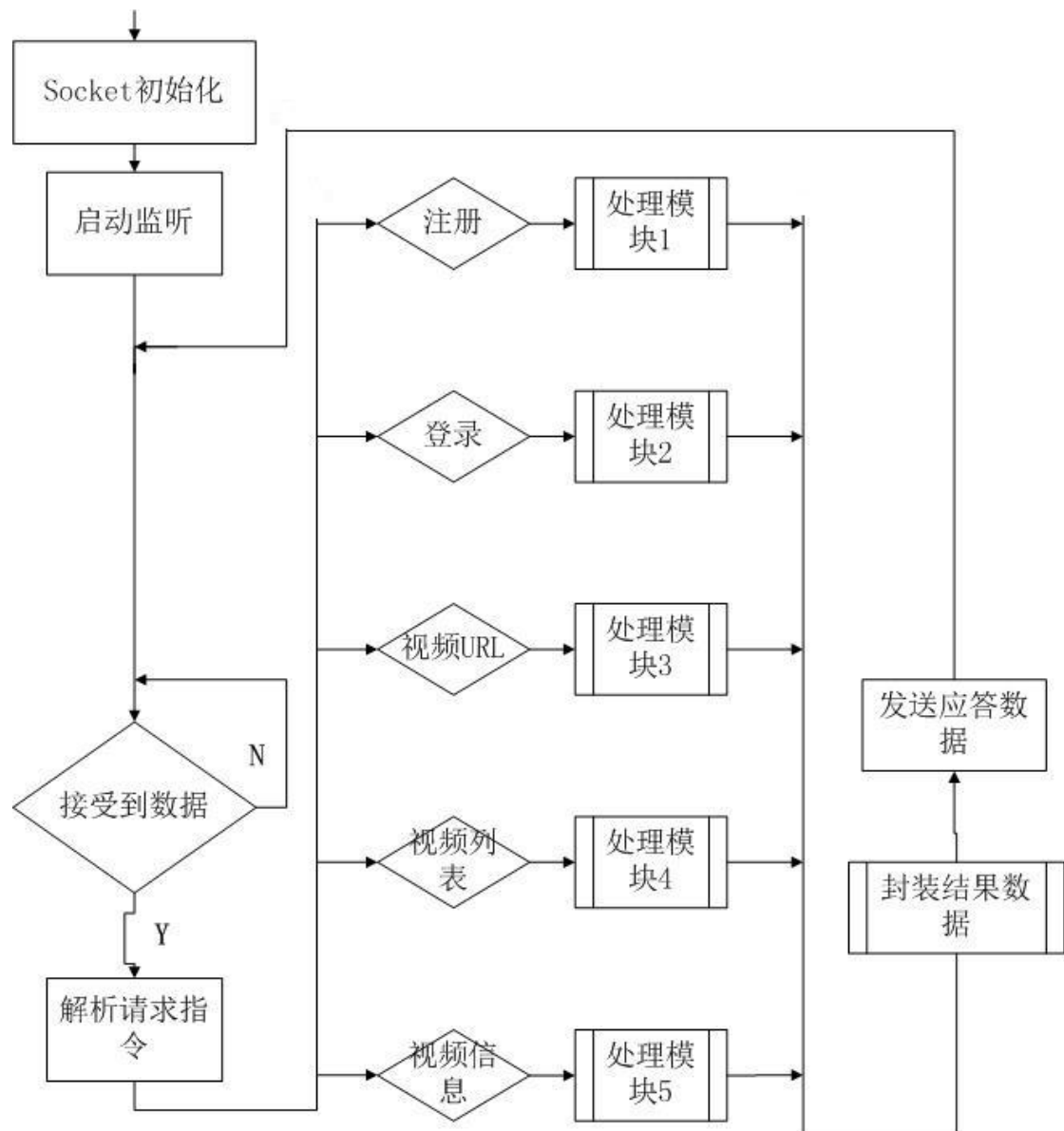


图 6-9 socket 服务器流程图

图 6-9 中的五个处理模块的处理流程是相似的，首先连接数据库，然后根据请求数据包中的参数进行数据操作，如添加、删除、改动数据库中的数据，最后对获得的数据进行处理。示意图如图 6-10 所示。

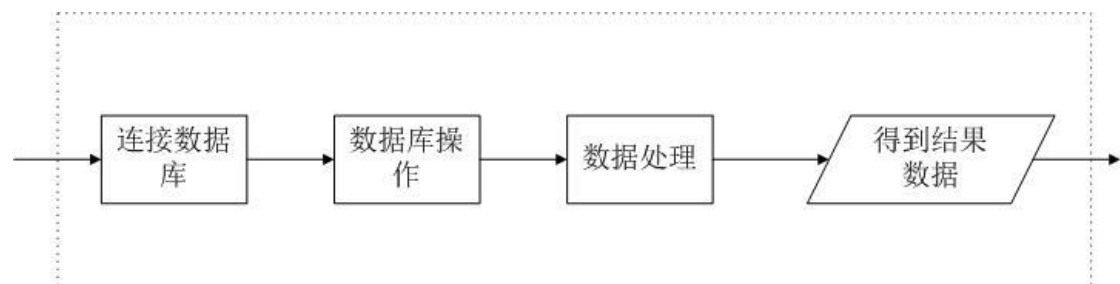


图 6-10 数据库相关的处理模块流程

封装结果数据这个处理模块的输出是要发给客户端的应答数据包，它封装了应答状态和结果数据。

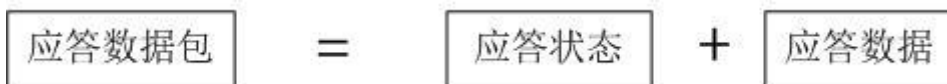


图 6-11 应答数据包封装示意图

6.3 移动客户端和 socket 服务器端的协议

从上一章移动客户端和 socket 服务器的规约一节的内容可以知道前者要向后者发送不同的指令，还要接收后者返回的数据。为便于二者对传送的指令和数据进行封装和解析，比如哪个字段是指令、哪个字段是数据以及是什么数据，我们需要对要传送的数据包的格式进行统一，即制定一套协议。

6.3.1 移动客户端向 socket 服务器端发送的请求指令的格式

- ①操作指令和参数都称作字段，各字段以符号“？”作分隔符
- ②每个数据包内的操作指令只有 1 条，假设参数的个数对相应执行模块时可知的，故数据包内没有对参数个数的说明
- ③首字段为操作指令，后面若干字段为参

操作指令	?	参数 1	?	参数 2	?	...
------	---	------	---	------	---	-----

图 6-12 请求指令

当 socket 服务器接受到一个请求数据包时，先对第一个字段进行解析从而获取操作指令，然后根据操作指令进入响应的处理模块，由相应的处理模块解析操作指令后面的参数。

6.3.2 socket 服务器端向移动客户端发送的应答数据的格式

- ①应答状态和应答数据都称作字段，各字段以符号“？”作分隔符
- ②每个数据包内的应答状态只有 1 条，假设应答数据的个数对相应执行模块时可知的，故数据包内没有对应答数据个数的说明
- ③首字段为应答状态，后面若干字段为应答数据

应答状态	?	数据 1	?	数据 2	?	...
------	---	------	---	------	---	-----

图 6-13 应答数据

移动客户端接收到 socket 的应答数据包时，先解析第一个字段从而获取应答状态，如果是成功的状态就解析应答数据。

6.3.3 协议的具体内容

表 6-7 是移动客户端和 socket 服务器之间五种操作过程的请求/应答数据包内部的字段定义，这五种过程依次是注册信息写入、登录信息认证、请求视频列表、请求视频 URL、请求某视频信息。

表 6-7 socket 协议内容

操作过程	划分	首字段内容	其余字段内容
注册 信息写入	请求	Register	四个参数，依次是用户名、密码、昵称和邮箱
	应答	RegisterOK	空
登录 信息认证	请求	Login	两个参数，依次是用户名、密码
	应答	LoginOK	空
请求 视频列表	请求	AllSong	空
	应答	AllSongOK	每个字段表示一首歌曲名称
请求 视频 URL	请求	URL	有一个参数，为视频名称
	应答	URLOK	所请求的视频的 URL
请求某视 频信息	请求	VideoInfo	有一个参数，为视频名称
	应答	VideoInfoOK	四个字段，依次是视频名称、作者、视频简介和 URL

6.4 流媒体服务器的布置

6.4.1 安装 DSS 流视频服务器

下载 DarwinStreamingSrvr5.5.5-Linux，解压，进入解压后的目录，执行如下命令来安装 DSS 流视频服务器⁴：

```

:~# addgroup --system qtss #创建qtss分组
:~# adduser --system --no-create-home --ingroup qtss qtss #创建qtss用户
:~# cd /home/now/DarwinStreamingSrvrlinux-Linux #进入安装目录
:/home/now/DarwinStreamingSrvrlinux-Linux# ./Install #执行安装

```

成功安装 DDS 流媒体服务器后，打开 IE 地址栏输入 <http://127.0.0.1:1220>，除了输入用户名和密码需要设置外，其余选项一律取默认值。

安装之后是默认启动的，如果服务器重启则必须重新启动流视频服务器和

⁴ <http://ubuntuforums.org/showthread.php?t=651556>

web 管理器，步骤如下：

启动 Darwin Streaming Server

```
:~# DarwinStreamingServer
```

启动 Web 管理介面(tcp port 1220)

```
:~# streamingadminserver.pl
```

6.4.2 部署视频文件

一般的 MP4 文件没有被流化，如果要被 darwin stream server 支持，必须对该视频文件添加 Hint 轨道。Hint 轨道之间互相关联，就像视频文件中音频轨道和视频轨道相互关联一样。Hint 轨道是 MP4 文件容器中的一系列的指令，它告诉流服务器怎么分发视频报文分组。例如发送分组的时机、要加上什么样的 RTP 头、分组里是否有外来数据、以及分组里含有多少数据等。

Linux 环境下用 MP4Box 程序进行预处理，步骤如下：

```
:~# apt-get install gpac
```

```
:~# MP4Box -hint 千千阙歌.mp4
```

然后把处理后的视频文件复制到/usr/local/movies，流视频服务器就布置好了。

6.4.3 流媒体服务器的测试

布置好流媒体服务器之后，需要对其进行测试，以确保其能正常运行。我们选择安装在同一局域网的另一台主机上的 VLC 播放器对流媒体服务器进行测试。VLC 多媒体播放器(全名是 VLC media player)是 VideoLAN 计划的多媒体播放器。它支持众多音频与视频解码器及文件格式，并支持 DVD 影音光盘，VCD 影音光盘及各类流式协议，其中就包括 RTSP 协议的流视频。



图 6-14 输入视频的 URL

图 6-14，图 6-15 是测试截图，可以看到能够正常播放服务器上的视频。



图 6-15 流视频播放场景

6.5 数据库的设计

要完成系统的功能我们需要两张数据库表，分别是视频信息数据表，用来存储视频信息，如视频名称、URL、作者和详细信息；用户信息数据表，用来储存用户名、密码、昵称和邮箱信息。

数据库 moviedatabase 如图 7-10 所示：



图 6-16 数据库 moviedatabase

6.5.1 视频信息数据表(streamMovieList)

表 6- 8 视频信息数据库表

字段名	类型	是否主键	默认值	附加属性	字段含义
ID	int(11)	PRIMARY	NULL	auto_increment	视频编号
name	varchar(50)	UNIQUE	NULL		视频名称
author	varchar(50)		NULL		作者
description	text		NULL		视频描述信息
URL	varchar(100)		NULL		视频的URL地址

视频编号是插入数据时系统按递增的顺序自动生成的，用作视频信息数据表的主键；视频名称是唯一的，不允许重复；视频描述信息是文本类型的，它一般内容较长，长短各异，故设置为文本类型；视频的 URL 信息是流视频的统一资源定位符，如 rtsp://10.109.16.148/千千阙歌.mp4。

6.5.2 用户信息数据表(userInfoList)

表 6-9 用户信息数据库表

字段名	类型	是否主键	默认值	附加属性	字段含义
ID	int(11)	PRIMARY	NULL	auto_increment	用户编号
user	varchar(50)		NULL		用户名
code	varchar(50)		NULL		密码
nickname	varchar(100)		NULL		昵称
email	varchar(100)		NULL		电子邮箱

用户编号是系统自动生成的，用作主键；用户名、密码用于用户登录验证。

数据库 moviedatabase 的两张数据表 userInfoList 和 streamMovieList 如图 6-17 所示：

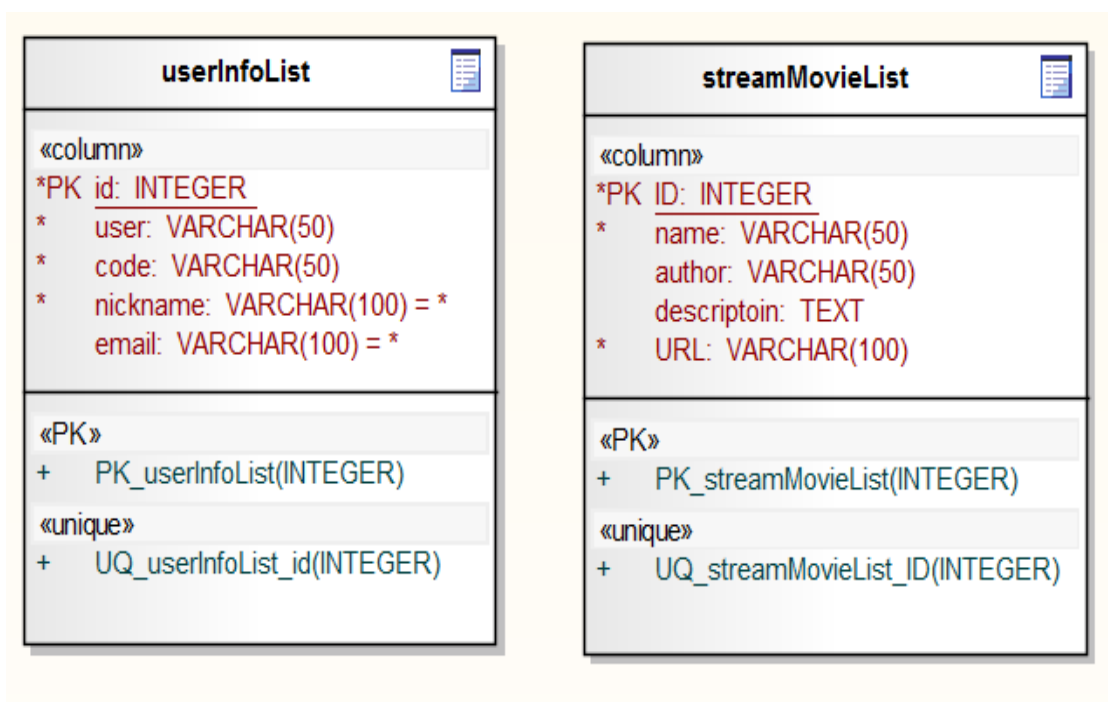


图 6-17 两张数据表

第七章 系统测试

系统各个模块功能实现后，我们将它们分别部署到相应的机器上进行性能测试。MySQL 数据库和 socket 服务器部署在同一台安装有 Ubuntu 系统主机上，流媒体服务器 Darwin Streaming Server 部署在另一台安装了 Ubuntu 系统的主机上，Android 客户端软件安装在 Android4.0 系统的一款三星手机上，发现系统能正常运行，各个功能都有很好地实现。下面依次从 Android 客户端、socket 服务器端和数据库端来展示系统运行情况。

7.1 Android 客户端的测试

Android 客户端有四个界面，分别是注册界面、登录界面、视频选择界面和视频播放界面。图 7-1 是 Android 的四个界面，按从左到右、从上到下的顺序依次是注册界面、登录界面、视频选择界面和视频播放界面。

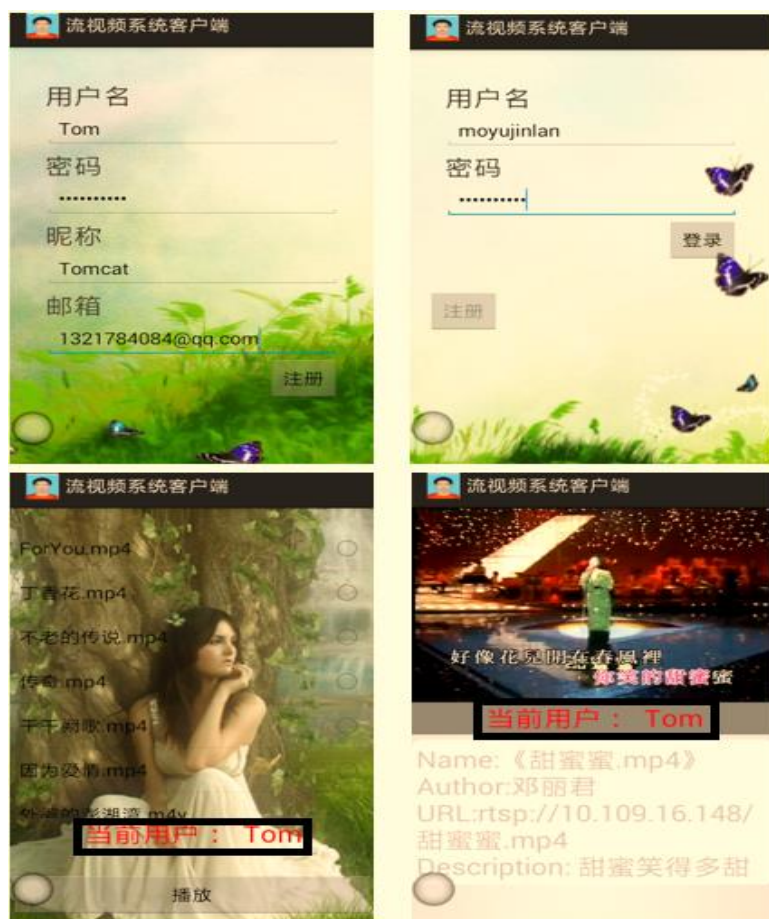


图 7-1 Android 客户端 UI 展示图

请注意图 7-1 中下半部分的两张 UI 截图中用粗线框标记的当前用户，说明客户端一直都保持着当前登录用户的状态。

在四个 UI 界面都可以查看客户端软件及作者的信息，此过程截图如图 7-2 所示。

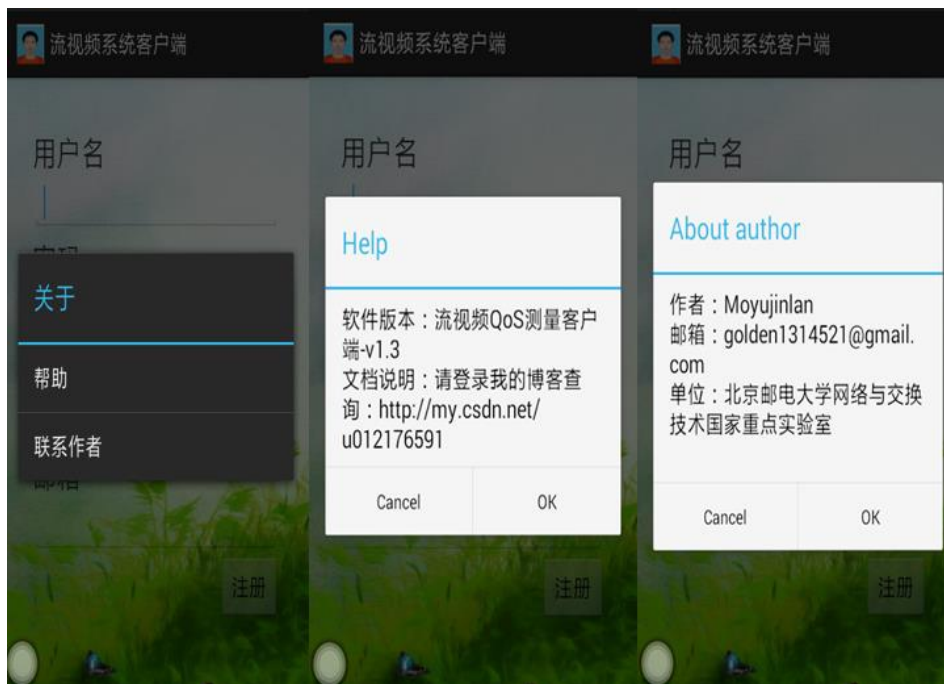


图 7-2 查看软件和作者信息

视频播放界面中显示视频信息的子界面能动态刷新字体颜色，这里使用了多线程机制。具体地说，主线程开启一个子线程负责视频信息的刷新工作，这个子线程调用了另一个类，而这个类调用了 C 语言编写的动态链接库。也就是说，这个功能的实现，是多线程机制和 NDK 开发相互配合的结果。



图 7-3 颜色动态刷新的效果图

图 7-3 是动态刷新效果：

7.2 socket 服务器端的运行记录

为了能够方便地从 socket 服务器运行记录中看到客户端和 socket 服务器端的信息交互过程，我们将 socket 服务器每次与客户端交互过程中从用户得到的请求指令数据包和向客户端发送的应答数据包的内容打印出来。数据包各字段以符号“?”为分隔符，各字段具体定义见第五章第 3 小节。

① 编译源码，得可执行文件 server.o：

```
root@jin-ThinkPad-R400:~# gcc $(mysql_config --cflags) server.c -o server.o $(mysql_config --libs)
root@jin-ThinkPad-R400:~#
```

② 启动 socket 服务器，从图 7-4 可以看到 socket 服务器启动的端口号：

```
root@jin-ThinkPad-R400:/usr/bin# server.o
Server started,port:34952
```

图 7-4 启动 socket 服务器

③图 7-5 是已注册用户的登录过程记录。

可以看到请求数据包的第一个字段是“Login”，表示登录，第二、三字段表示用户名和密码。应答数据包包含 5 个字段，第一字段是“LoginOK”，表示应答成功，其余字段表示用户信息，如用户名、密码、昵称、邮箱。

```
*****
request from client : Login?jin?dog
write to client : LoginOK?jin?n?n?fish can fly?jobs8888168@baidu.com$
*****
```

图 7-5 用户登录

④图 7-6 是用户注册过程。

可以看到请求数据包的第一字段是“Register”，表示注册，后面四个字段是注册用户提交的个人信息，依次是用户名、密码、昵称和电子邮箱。应答数据包只有一个字段，内容是“RegisterOK”，表示应答成功。

```
*****
request from client : Register?Tom ?abcd123456?Tomcat ?1321784084@qq.com
INSERT INTO userInfoList(user,code,nickname,email) VALUES('Tom ','abcd123456','Tomcat ','1321784084@qq.com')
write to client : RegisterOK?$
*****
```

图 7-6 用户注册

⑤图 7-7 是客户端请求视频名称列表的过程。

可以看到请求数据包只有一个字段，内容是“AllSong”，表示请求视频列表。应答数据包的内容是以符号“？”为间隔符的一系列视频名称。

```
*****
request from client : AllSong
write to client : ForYou.mp4?丁香花.mp4?不老的传说.mp4?传奇.mp4?千千阙歌.mp4?因为爱情.mp4?外婆的澎湖湾.m4v?心中的日月.mp4?星星.mp4?最浪漫的事.mp4?樱花.mp4?歌剧2.mp4?猫和老鼠.mp4?甜蜜蜜.mp4?菊花台.mp4?鸟儿.mp4?$
*****
```

图 7-7 获取视频列表

⑥图 7-8 是请求某视频的 URL 的过程。

可以看到请求数据包的第一字段是“URL”，表示请求视频的 URL，第二个字段是要请求 URL 的视频的名称，服务器将根据本字段的内容向数据库查询响应的 URL。应答数据包仅有一个字段，是与请求数据包中的视频名称相对应的 URL。

```
*****
request from client : URL?传奇.mp4
write to client : rtsp://10.109.16.148/传奇.mp4?$
*****
```

图 7-8 请求 URL

⑦图 7-9 是请求某视频所有信息的过程。

请求数据包的第一字段是“VideoInfo”，表示要查询视频的所有信息，第二字段是响应的视频的 URL，服务器以本字段的内容确定视频对象。应答数据包有五个字段，第一字段表示应答状态，后面四个字段表示视频的详细信息，依次是视频名称、作者、视频描述信息、视频 URL。

```
*****
request from client : VideoInfo?rtsp://10.109.16.148/传奇.mp4
write to client : InfoOK?传奇.mp4?王菲?歌曲《传奇》被王菲唱红?rtsp://10.109.16.148/传奇.mp4?$
*****
```

图 7-9 请求某个视频所有信息

7.3 数据库端的测试

我们新建了一个数据库 moviedatabase，其内部包含两张数据表，分别是 userInfoList, 用来存储用户信息；streamMovieList, 用来存储视频信息。

下面进入 MySQL 数据库软件的命令行操作界面下查看我们所设计的数据库信息。

图 7-12 是数据表 userInfoList 的属性描述。

```
mysql> desc userInfoList;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
user	varchar(50)	NO		NULL	
code	varchar(50)	NO		NULL	
nickname	varchar(100)	NO		*	
email	varchar(100)	NO		*	

5 rows in set (0.00 sec)

图 7-10 命令行下查看 userInfoList 属性

图 7-13 是数据表 userInfoList 存储的用户信息，可以看到目前数据库中有三个注册用户，我们可以看到它们的用户 ID、用户名、登录密码、昵称和电子邮件。

```
mysql> select * from userInfoList;
```

id	user	code	nickname	email
1	moyujinlan	abcd123456	fish can fly	jobs8888168@baidu.com
2	goldenfather	jobs1991	happy sunday	golden1314521@gmail.com
3	jin	dog	foolish,hungry	1321784084@qq.com

3 rows in set (0.00 sec)

图 7-11 命令行下查看 userInfoList 数据

图 7-14 是数据表 streamMovieList 的属性描述。

```
mysql> desc streamMovieList;
```

Field	Type	Null	Key	Default	Extra
ID	int(11)	NO	PRI	NULL	auto_increment
name	varchar(50)	NO	UNI	NULL	
author	varchar(50)	YES		NULL	
description	text	YES		NULL	
URL	varchar(100)	NO		*	

```
5 rows in set (0.02 sec)
```

图 7-12 命令行下查看 streamMovieList 属性

图 7-15 是数据表 streamMovieList 存储的视频信息。由于字段 description 存储的文本信息比较长，不易观察，我们这里只查看其余四个字段的信息，包括 ID、视频名称、作者和视频的 URL 信息。

```
mysql> select ID,name,author,URL from streamMovieList;
```

ID	name	author	URL
2	樱花.mp4	NO	rtsp://10.109.16.148/樱花.mp4
3	最浪漫的事.mp4	邓丽君	rtsp://10.109.16.148/最浪漫的事.mp4
35	不老的传说.mp4	张学友	rtsp://10.109.16.148/不老的传说.mp4
36	传奇.mp4	王菲	rtsp://10.109.16.148/传奇.mp4
37	丁香花.mp4	唐磊	rtsp://10.109.16.148/丁香花.mp4
38	歌剧2.mp4	Vitas	rtsp://10.109.16.148/歌剧2.mp4
39	菊花台.mp4	周杰伦	rtsp://10.109.16.148/菊花台.mp4
40	猫和老鼠.mp4	威廉·汉纳	rtsp://10.109.16.148/猫和老鼠.mp4
41	ForYou.mp4	NO	rtsp://10.109.16.148/ForYou.mp4
42	鸟儿.mp4	NO	rtsp://10.109.16.148/鸟儿.mp4
43	千千阙歌.mp4	张国荣	rtsp://10.109.16.148/千千阙歌.mp4
44	甜蜜蜜.mp4	邓丽君	rtsp://10.109.16.148/甜蜜蜜.mp4
45	外婆的澎湖湾.m4v	潘安邦	rtsp://10.109.16.148/外婆的澎湖湾.m4v
46	心中的日月.mp4	王力宏	rtsp://10.109.16.148/心中的日月.mp4
47	星星.mp4	NO	rtsp://10.109.16.148/星星.mp4
48	因为爱情.mp4	李亚鹏	rtsp://10.109.16.148/因为爱情.mp4

```
16 rows in set (0.00 sec)
```

图 7-13 命令行下查看 streamMovieList 数据

7.4 流媒体服务器的运转

图 7-16 是流媒体服务器 Darwin Streaming Server 的运行状态参数表。

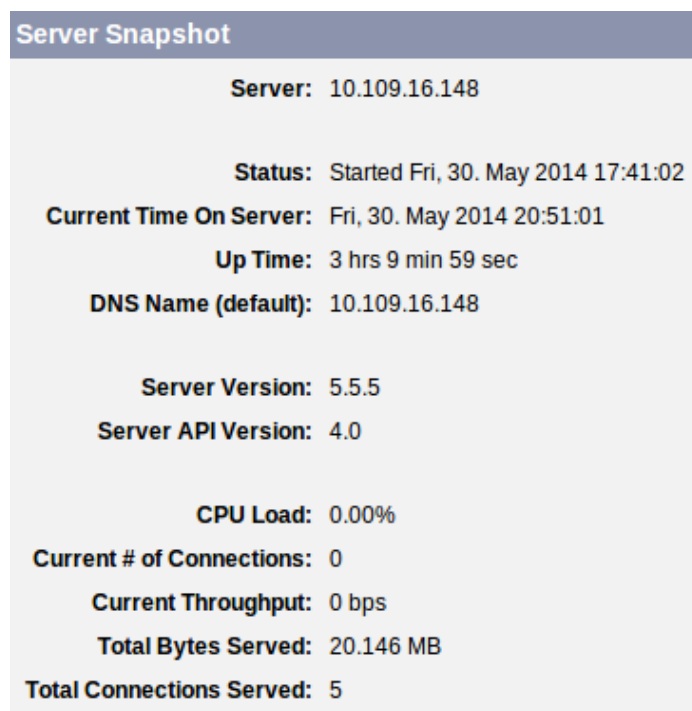


图 7-14 DSS 运行参数

图 7-17 是流视频在服务器端的目录。

```
root@jin-ThinkPad-R400:~# cd /usr/local/movies/
root@jin-ThinkPad-R400:/usr/local/movies# ls
ForYou.mp4          你是我心内的一首歌.mp4  星星.mp4            甜蜜蜜.mp4
sample_h264_300kbit.mp4  千千阙歌.mp4            最浪漫的事.mp4     菊花台.mp4
丁香花.mp4          因为爱情.mp4            樱花.mp4            鸟儿.mp4
不老的传说.mp4      外婆的澎湖湾.m4v        歌剧2.mp4
传奇.mp4             心中的日月.mp4          猫和老鼠.mp4
```

图 7-15 流视频存储目录

第八章 结束语

8.1 总结

移动通信和互联网成为当今世界发展最快、市场潜力最大、前景最诱人的两大业务。它们的增长速度都是任何预测家未曾预料到的。迄今，全球移动用户已超过 15 亿，互联网用户也已逾 7 亿。中国移动通信用户总数超过 3.6 亿，互联网用户总数则超过 1 亿。这一历史上从来没有过的高速增长现象反映了随着时代与技术的进步，人类对移动性和信息的需求急剧上升。越来越多的人希望在移动的过程中高速地接入互联网，获取急需的信息，完成要做的事情。所以，出现的移动与互联网相结合的趋势是历史的必然。移动互联网正逐渐渗透到人们生活、工作的各个领域，短信、铃声下载、移动音乐、手机游戏、视频应用、手机支付、位置服务等丰富多彩的移动互联网应用迅猛发展，正在深刻改变信息时代的社会生活，移动互联网经过几年的曲折前行，已经迎来了新的发展高潮。

如今，移动应用技术已经进入到千家万户，影响到几乎每一个社会人，可以说短短十几年依然彻底改变了整个世界的面貌和人类的生活方式。移动应用生态系统不仅极大地促进了人类生产力的发展，几倍甚至几十倍地提高了人类工作的效率，还在很大程度上方便了人类的生活，使我们的生活变得丰富多彩。智能手机、PAD、平板等移动终端设备已经成为人们的生活必需品，游戏、电影、电子邮件、移动支付等应用软件充斥着人们的生活。

随着 4G 时代的到来，数据传输速率大幅度提升，为移动互联网的发展提供了坚实的网络基础和广阔的发展空间。4G 是集 3G 与 WLAN 于一体，并能够快速传输数据、高质量、音频、视频和图像等。4G 能够以 100Mbps 以上的速度下载，比目前的家用宽带 ADSL（4M）快 20 倍，并能够满足几乎所有用户对于无线服务的要求。此外，4G 可以在 DSL 和有线电视调制解调器没有覆盖的地方部署，然后再扩展到整个地区。很明显，4G 有着不可比拟的优越性。

在这种情况下，移动流视频必将呈现爆炸式的增长。本文设计并实现了一套基于移动互联网的流媒体点播系统，包括 Android 客户端、流媒体服务器、socket 服务器和数据库，能够满足用户注册、登录、选择流视频节目、观看流视频和查看流视频信息等需求。具体而言，本文主要完成的工作及创新有：

(1) 根据需求将系统分为四大部分，分别是手机流媒体客户端、流媒体服务器、socket 服务器和数据库。

(2) 采用 RTSP 协议簇，音视频流数据采用 RTP/RTCP 协议传输。

(3) socket 服务器采用 C 语言编写，运行在 Ubuntu 操作系统上，是一个守护进程，负责接受手机客户端软件的 socket 客户端的服务请求。除此之外，它还负责与 MySQL 数据库交互。Socket 客户端程序用 Java 语言编写，与 C 语言编写

的 socket 服务器之间的沟通没有障碍。

(4) 用户注册时提交的用户信息会写入数据库的用户信息表中。视频信息的数据，特别是视频名称和 URL 信息存储在数据库的视频信息数据表中，由系统管理员维护。

(5) 流媒体服务器采用开放源码的 DDS 服务器，它支持 RTSP/RTCP/RTP 协议，而且由于其开源性，方便了我们以后的后续开发。

(6) Android 客户端采用了 Java 和 C 语言混合编程，具有清晰的类结构和层次，不但容易理解，而且如果我们以后对功能进行扩展的话，工程量将比较小。

8.2 展望与改进方向

随着移动互联网带宽的增加和单位流量价格的递减，上网会越来越便宜，网速越来越快。移动多媒体应用特别是视频应用具有广阔的发展前景，许多相关的视频业务将兴起，并很快重塑互联网服务行业的结构。这些业务包括。

1. 移动视频消息

移动视频消息是通过某种媒体非实时地发送视频消息，包括人到人、人到 PC 以及 PC 到人等三种情况。例如，目前正在推广的视频邮件和多媒体消息业务就属于移动视频消息业务。目前实现的移动视频消息业务中所传送的视频消息一般为视频动态图像。

(1) 视频邮件

视频邮件是在纯文本邮件的基础上，增加了视频和音频的多媒体邮件。视频邮件可以用附件形式传送，也可以直接以流媒体方式播放。视频邮件不仅使个人移动通信变得丰富多彩，也可以加强企业电子商务的广告效应。

(2) 多媒体消息

多媒体消息业务是一种被 WAP 组织和 3GPP 定义为标准的非实时移动消息业务，移动用户通过一个或多个媒体单元来发送和接收视频消息。多媒体消息业务不仅可传递文本信息，而且还可以传递内容更为丰富的图像、音频、视频等数据信息，将是 2.5G 和未来 3G 的一种核心业务。

2. 移动视频内容配送

移动视频内容配送业务主要是从服务器到人的通信服务，以流媒体或者下载的方式将视频内容传送到移动用户终端。移动视频配送的业务范围非常广泛，主要包括视频点播和远程教育业务等。流媒体与下载方式的主要区别在于，流媒体是指通过 IP 网络以流的方式传送多媒体内容（包括音频和视频），不需要下载整个文件。流是指数据从服务器端向客户端以不间断的流方式传送，当第一个数据包到达时就开始播放。

基于流媒体技术的远程教育为人们提供了一种新的学习方式，更适合社会发

展的需要。目前，远程教育还局限在室内电视或者 PC 机，这大大限制了人们的活动范围。而基于移动视频的移动远程教育将使人们随时随地的接收新知识，将工作、学习、休闲融为一体。

随着 Internet 和移动网络的融合，人们越来越希望能够及时、主动地获取各种视频节目，因此视频点播业务将极大地满足人们娱乐的需要。目前，全球的移动运营商正在逐步推出采用流媒体方式实现的视频点播业务，以吸引更多用户。

3. 移动视频游戏

目前，移动游戏的种类越来越多，移动游戏市场的竞争也越来越激烈。但是，嵌入在手机终端的游戏、SMS 游戏等画面单一，给用户带来的感受无法同交互式视频游戏相比。现在，基于 Java 和 Brew 的彩屏手机终端可以下载游戏，在某种程度上改变了移动游戏原有的单一面貌。但是，交互式网络游戏将是移动视频游戏发展的最终趋势，同单机游戏相比，网络游戏给移动运营商带来的收益更多。

4. 移动视频电话/会议

移动视频电话是一种使用了视频和话音的点对点通信业务，在两个移动终端、移动终端和固定视频电话或者 PC 机等之间实现视频和音频的双向实时交流。移动视频电话业务是由 UMTS 定义的一种高级 3G 业务，对无线网络带宽有较高的要求。带宽、成本、设备和服务的限制使这一业务的开展具有很大的难度。在宽带网络出现后，移动视频电话/会议才有可能以可以接受的质量被大规模推广。

移动视频电话/会议是比话音、数据通信更高级的一种通信形式，实现了点对多点的通信业务，极大地满足了人们个人通信和商务交流的需求，为商务用户带来更大的便利。现在，各国的 3G 移动运营商将移动视频电话业务作为 3G 业务拓展的先锋，和记在欧洲正大力推广移动视频业务。同时，日本 NTT DoCoMo 公司已实现了 3G 终端和 PC 间的视频会议业务。

5. 移动电子商务

社会的发展产生了多种消费方式，电子商务已走进了人们的生活。移动电子商务使用移动终端进行电子商务交易，是电子商务向移动通信领域的发展。移动视频与电子商务的结合，将使用户可以随时随地利用移动终端的视频业务了解商品信息，进行网上购物。

6. 移动视频监控

利用无线网络的高带宽，通过移动网络和移动、固定视频前端，可实现远距离的移动视频监控。一般情况下，移动视频监控业务用于特定的工作用途，也可以用于生活中，如用于儿童安全保护的移动监控系统。

移动视频监控系统可以采用基于 IP 传输的视频编码设备，通过无线网络将视频信号传送到中心存储和显示。远程实时监控业务除了在社区推广，也能扩展到幼教领域，家长通过移动终端上网就可以实时观看到孩子在幼儿园的生活情况。

不同的联网方式（如 2G、3G、WiFi 等）的带宽、覆盖范围、工作原理各不相同，使移动终端获得的流视频的 QoS（如时延、抖动、丢包率、误码率等）有明

显的差别。因而当用户面对不同的联网费用和不同的用户体验时将有所选择，所以为不同的联网方式制定合理的价格将对移动互联网的发展起着重要的促进的作用。价格的确定，最重要的参数就是服务质量。

因此，下一步的研究目标就是获得移动终端在不同联网方式下的服务质量即 QoS 的数据，对其进行分析，并分别形成 2G、3G、WiFi 的 QoS 的一般标准，为移动互联网的数据业务提供行业定价标准。

参考文献

- [1] 温 伟, 刘荣科. Android 多媒体框架下 Stagefright 的功能扩展. 太赫兹科学与电子信息学报. 11 (5). 2013. 12: 718-723.
- [2] 杨长刚. 深入剖析 Android 系统. 第 1 版. 电子工业出版社. 2013: 331-355.
- [3] Martin Fowler. UML 精粹 :标准对象建模语言简明指南. 第 3 版. 北京: 电子工业出版社. 2012. 6: 43-77.
- [4] 王家林, 王家俊, 王家虎. Android 高级开发实战:UI、NDK 与安全. 第 1 版. 北京: 电子工业出版社. 2013: 240-337.
- [5] 谢希仁. 计算机网络. 第 5 版. 北京: 电子工业出版社:2010:306-322.
- [6] 蔡庆元. 基于 RTP 的手机视频点播系统的设计与实现. [学位论文]. 华南理工大学. 2010
- [7] 肖芳杨. 基于 Android 平台的移动实时监控系统设计. [学位论文]. 江西理工大学. 2012.
- [8] 林增. 浅析流媒体技术及其应用的主流解决方案. 计算机光盘软件与应用 . (3). 2010: 89-90.
- [9] 王远洋. Linux 下基于 socket 多线程并发通信的实现. 微计算机信息. 25 (5-3). 2009. 5: 70-72.
- [10] 鄂大伟. 软件工程. 第 1 版. 北京: 清华大学出版社. 2010. 8: 312-367.

致 谢

在做毕业设计的三个月中，我涉猎到了好多东西，有时甚至需要对相关的某个方向做深入研究，这一切使我进一步学习了系统的专业知识，开阔了思考和学习的领域和视野，锻炼了我实际解决问题的能力和进行科学研究的思维方法。我深知一切的一切离不开指导老师和同学们的指导和帮助，是这些良师益友有形无形的督促和帮助才使我快速成长。

感谢我的指导老师王尚广副教授，感谢他的对我的严格要求、指导和一直以来的支持，使我能够全身心地投入毕业设计的工作，而没有感到厌倦。

感谢孙其博副教授，感谢他一直以来对我的提醒和关怀，这使我能够从容地安排我的工作。

感谢李硕师兄不厌其烦对我的不厌教导，每当我在没有思路的时候，他总是能够使我茅塞顿开，使我能够一直保持昂扬的斗志。

感谢实验室的其他几位同学，他们就像我的战友，我们一起工作，互相激励，互相帮助，与他们在一块儿的时光使我感到非常幸运。

最后感谢我自己，感谢我一直以来的艰苦卓绝的努力，感谢我屡败屡战、精益求精的精神，没有这些，所有的都无从谈起。