

基于 Android 系统的影音播放器开发

张正政, 林耀荣

(华南理工大学 电子与信息学院, 广东 广州 510640)

摘要: 基于开源移动终端操作系统 Android 开发了一个影音播放器。影音播放器由文件浏览器、视频播放、音频播放、播放列表管理及歌词同步显示等模块组成, 界面采用 XML 脚本配置生成。软件基于文件浏览器的方式实现了媒体库的管理, 大大方便了用户的操作。软件每次启动时都会对储存设备进行扫描, 将扫描到的歌词文件信息保存到歌词数据库中。对正在播放的歌曲, 根据歌曲名字检索歌词文件, 并同步显示歌词。在 Android 模拟器和 OMAP 3530 开发板上对播放器进行了功能测试与验证。

关键词: Android; 影音播放器; 歌词显示; OMAP3530

中图分类号: TN911-34; TP316.2

文献标识码: A

文章编号: 1004-373X(2011)02-0005-04

Development of Android-based Mediaplayer

ZHANG Zheng-zheng, LIN Yao-rong

(School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510640, China)

Abstract: The Android-based mediaplayer is developed. The mediaplayer is composed of file browser, video and audio playback, playlist management and lyric display modules, etc. The media on mediaplayer are managed by file browser to facilitate the operation. The lyric files on the devices' storage media are searched and the lyric database is updated while the software is launched. When playing music, the software will search the lyric database for lyric display. The mediaplayer is tested in Android emulator and finally ported to OMAP3530 development board.

Keywords: Android; mediaplayer; Lyric display; OMAP3530

0 引言

Android 一词的本义指“机器人”, 同时也是 Google 于 2007 年 11 月 5 日宣布的基于 Linux 平台的开源移动终端操作系统的名称, 该平台由操作系统、中间件、用户界面和应用软件组成, 是首个为移动终端打造的真正开放和完整的移动软件^[1]。

随着科技的不断发展, 手机不再只是人们日常生活的移动通信工具, 而是一个多媒体平台, 可提供强大的多媒体功能, 越来越多的人喜欢用手机听音乐、看电影。Android 系统内置的音乐播放器基于媒体库对歌曲进行管理, 并且不支持歌词的同步显示。对于部分已经习惯的基于文件浏览器形式播放器的用户, 操作起来会有所不便。对于 Android 手机操作系统, 可以充分利用它丰富的可编程扩展性, 开发出新的播放器应用软件。

1 Android 平台

Android 软件开发主要涉及到的相关技术包括 Android SDK(开发插件、调试工具包等)、Android 系统架构和 Android 应用程序组件。

1.1 Android SDK

Android SDK(Software Development Kit)提供了 Android 平台上使用 Java 语言进行 Android 应用开发的 API 接口和工具。其中, 最重要的工具是 Android 模拟器和 Eclipse^[2] 的 Android 开发工具插件, 同时 SDK 也包含了各种在模拟器上用于调试、打包和安装的工具。

Eclipse IDE 的 Android 开发工具插件(Android Development Tools Plugin for the Eclipse IDE, ADT)大大扩展了 Eclipse 集成环境功能, 使得生成和调试 Android 应用程序既容易又迅速。使用 Eclipse, ADT 插件可以加快开发 Android 应用程序的速度。

Android 模拟器是在计算机上运行的一个虚拟移动设备, 可以使用模拟器来模拟一个实际的 Android 运行环境, 来设计、调试和测试应用程序。

收稿日期: 2010-08-18

基金项目: 广东省自然科学基金资助项目(9351064101000003);

华南理工大学中央高校基本科研业务费专项资金资助
(2009ZM0104)

Dalvik 调试监视器服务 (Dalvik Debug Monitor Service, DDMS) 集成了 Dalvik (为 Android 平台定制的虚拟机 (VM)), 能够在模拟器或者设备上管理进程并协助调试。DDMS 提供进程管理, 产生跟踪数据, 观察堆和线程信息, 截取模拟器屏幕画面, 模拟来电或短信等功能。

1.2 Android 系统架构

Android 是一个专门针对移动设备的软件集, 它包括一个操作系统, 中间件和一些重要的应用程序^[3]。Android 的系统架构从上往下依次是应用程序、应用程序框架和组件、Android 运行时库和其他库、优化的 Linux 内核^[4-5]。

1.3 Android 应用程序的组件

手机的运行环境比较复杂, 一个程序运行时经常会出现很多的突发情况, 例如, 来电或者短信及电量不足等。这些问题在 Android 中都有解决方案, 在 Android 的应用程序中, 以下几个组件特别重要:

(1) Activity

顾名思义, Activity 就是活动。应用程序必须至少包含一个 Activity。Activity 提供了一个虚拟的用户接口, 每一个 Activity 都有一个默认的窗口可以用来显示用户界面, 一般情况下, 它是全屏的窗口。例如, 影音播放软件的每个用户界面就是 Activity。

(2) Service

Service 与 Activity 的主要区别在于, Service 是在后台活动的, 它没有用户界面^[6]。典型的应用场景是, 用户选择了一个播放列表开始播放后, 离开了该用户界面 (Activity), 如果仍然希望歌曲不会暂停, 就可以用 Service 去调用播放器, 而不是在 Activity 中调用播放器。

(3) Intent

在 Android 中, 每一个行为都可以抽象为一个 Intent。可以简单地把 Intent 理解为一个包含了数据的信息。Intent 可以用于启动一个 Activity 或者 Service, 可以用于进程间的通信。事实上, 应用程序的启动也是通过 Intent 实现的。

(4) Content providers

Android 平台内置了一套 SQLite 数据储存机制, 并包含一系列管理 SQLite Database 的相关方法^[7]。在应用程序中, 可以通过 Content providers 访问数据库。每个 Content provider 都定义了一系列的方法用于访问它对应的数据库。应用程序在新建数据库时也可以定义相应的 Content provider 用于向其他应用程序共享数据。一般情况下, 应用程序不直接调用 Content provider 定义的函数, 而是通过 ContentResolver

间接调用。这样设计的好处是一个 ContentResolver 可以访问任何的 Content provider, 统一了接口。

2 硬件平台

系统的硬件平台采用基于 OMAP3530 的开发板 DevKit8000。美国 TI 公司的 OMAP3530 应用处理器^[8]集成了 600 MHz 的 ARM Cortex-A8 CPU 内核和 430 MHz 的 C64x+ DSP 内核, 并提供了 USB 2.0 OTG, SD/MMC 和 LCD 等丰富的外设接口, 适用于各种无线手持终端应用。DevKit8000 开发板外接了一个 4.3 寸的 LCD 屏、电阻式触摸屏和音频输入/输出接口, 可以用于对基于 Android 系统的影音播放器的开发。

通过打补丁包和开发 LCD 驱动、触摸屏驱动、按键驱动等外设驱动, 在 DevKit8000 开发板上移植了版本为 6.2.29 的 Linux 内核和版本为 1.6 的 Android 系统。

3 Android 影音播放器软件开发

3.1 软件功能

Android 的影音播放软件主要包括以下功能:

(1) 播放视频

支持快进、快退、暂停、继续和进度拖拽。

(2) 播放音频

支持快进、快退、上一首、下一首、暂停、继续、停止和进度拖拽; 支持随机播放、循环播放两种模式; 支持歌词同步显示。

3.2 各模块之间的关系

影音播放软件工程文件主要包括 src 文件夹 (Java 源码)、res 文件夹 (资源文件) 和 AndroidManifest.xml (程序清单)。软件主要由 8 个 Activities 组成, 每个 Activity 对应于一个模块。不同的 Activities 通过 Intent 进行通信和相互调用。

各个模块之间的关系如图 1 所示。

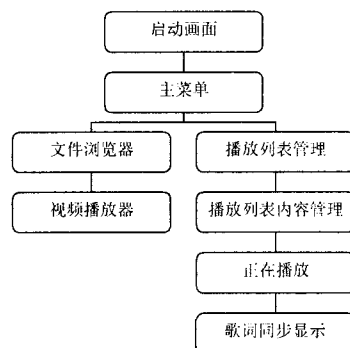


图 1 各个模块之间关系

视频播放调用顺序依次是<启动画面>、<主菜单>、<文件浏览器>、<视频播放器>;

音频播放调用顺序依次是<启动画面>、<主菜单>、<播放列表管理>、<播放列表内容管理>、<正在播放>。在<正在播放>的界面上,点击相应的按钮就可以执行歌词同步显示功能。

在工程目录的 AndroidManifest.xml(清单文件)里面,添加各 Activity 的说明。下面为对应启动画面的描述:

```
//Activity 的名字(Splash)
<activity android:name=". Splash"
    android:label="@string/app_name">
    <intent-filter>
    //感兴趣的 Intent
    <action android:name="android.intent.action.
MAIN" />
    //收到该 Intent 后采取的操作方式,LAUNCHER
表示它是程序入口点
    <category
    android:name="android.intent.category.
LAUNCHER" />
    </intent-filter>
</activity>
...
```

3.3 软件模块

根据影音播放软件功能,可以将软件划分为以下几个模块:

(1) <启动画面>模块。主要功能是显示软件启动画面。利用定时器线程去控制启动画面显示时间,并且当用户按下按键的时候,会立即中断启动画面的显示并跳转到<主菜单>模块。

(2) <主菜单>模块。主要功能是提供播放软件的用户界面。用户界面由 Android 的 XML 配置生成。主菜单提供三个选项,分别是视频播放、音频播放和退出程序;视频播放时,调用<文件浏览器>模块,让用户选择需要播放的视频文件。音频播放时,调用<播放列表管理>模块,让用户选择需要播放的列表。

(3) <文件浏览器>模块。用于浏览内存卡上的文件。如果路径是文件夹,则列出该文件夹下的所有文件;如果路径是文件,则先检查文件扩展名是否为支持的视频格式,如果是,则调用系统内置的播放器播放。

(4) <音频播放>模块。它包括以下几个模块:

① <播放列表管理>模块。主要功能是添加、删除、重命名列表。通过 Content providers 访问系统的播放列表数据库,并利用修改的结果更新数据库。因为采用的是系统的数据库,所以与系统内置的音乐播放器列表共用。同时,它还提供添加列表内容的接口。通过播放列表管理模块进入到<播放列表内容管理>模块。

② <播放列表内容管理>模块。主要功能是添

加、删除、多选播放列表中的歌曲。在 Android 中,每个列表都会有一个相应的列表内容库,保存了该列表里面所包含的歌曲 ID。通过 Content Providers 可以访问并修改。其中,添加功能是基于文件浏览器实现的,在文件浏览器中可以对音乐文件的进行标记和添加。

③ <正在播放>模块。主要功能是显示歌曲的相关信息(专辑封面、专辑名称、歌手、长度等)和显示正在播放歌曲的时间进度,控制歌曲的播放。

<正在播放>模块的界面用 XML 配置生成。

```
//总体上是一个线性的布局
//布局的宽度:填充父窗口 高度:能容纳下子控件 线性排
列方式:水平
<LinearLayout android:layout_width="fill_parent"
    android:layout_height="wrap_content" android:orien-
tation="horizontal"
    android:paddingTop="3px">
    //专辑画面的控件说明
    <ImageView android:layout_width="200px"
    android:layout_height="200px" android:layout_
marginLeft="4dip"
    android:layout_marginRight="4dip" android:scale-
Type="fitXY"
    android:src="@drawable/ic_mp_albumart_un-
known"
    android:id="@+id/albumview" />
    //其他的一些控件说明
    ...
</LinearLayout>
```

实际使用的效果图如图 2 所示。

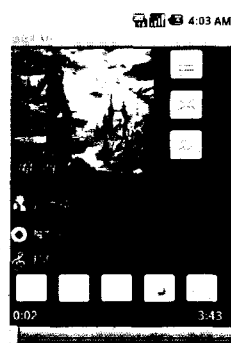


图2 播放界面

在 Android 的音乐库已经将歌曲的 Idv2 或 Idv3 标签^[9]进行解析,并将歌曲的相关信息(包括演唱者、流派、时长等)保存在媒体数据库中。只需要通过 Content providers 对该数据库进行访问,就可以得到相关的信息并送显示。获取歌曲相关信息的程序源码如下:

```
//获取歌曲信息的条目
Cursor cr = managedQuery(now_playing_uri, null,
null, null, null);
//显示歌曲的演唱者,专辑信息,歌曲名字
artistname.setText(cr.getString(cr.getColumnIndex-
OrThrow(Media.ARTIST)));
albumname.setText(cr.getString(cr.getColumnIndex-
OrThrow(Media.ALBUM)));
trackname.setText(cr.getString(cr.getColumnIndex-
OrThrow(Media.TITLE)));
```

```

//获取专辑的 ID
Long ablum_id = cr.getLong(cr.getColumnIndex
(Media, ALBUM_ID));
//从专辑库中检索出该专辑信息的条目
Uri albums_art =
//这个函数是生成专辑的检索地址
//第一个参数是由 Content Provider 提供的数据库入口
地址
//第二个参数是专辑的 ID
ContentUris.withAppendedId(Audio, Albums, EXTERNAL
_CONTENT_URI, ablum_id);
//检索专辑
Cursor ablum = managedQuery(albums_art, null, null,
null, null);
//获取该专辑封面的路径
String path = ablum.getString(ablum.getColumnIndex
(Audio, Albums, ALBUM_ART));
//显示专辑封面, albumview 是一个 ImageView 空间
albumview.setImageURI(Uri.parse(path));

```

通过创建 MediaPlayer 播放器类对象,调用 Android 提供的 API 就可以实现文件的解码和播放。

④ <歌词同步显示>模块。程序启动时,对内存卡进行扫描,将扫描到的歌词文件名字和路径保存到歌词数据库中。对正在播放的歌曲,根据歌曲名字自动检索歌词文件,并同步显示歌词。第一次启动该程序时,需要新建歌词数据库,用于保存歌词文件信息。Android 提供了类 SQLiteOpenHelper 用于新建数据库。

```

public void onCreate(SQLiteDatabase db)
{
//用 SQL 原语新建歌词数据库
db.execSQL(DATABASE_CREATE);
}

```

DATABASE_CREATE 是一个 SQL 原语的字符串宏,其内容是“create table lyric_meta (_id integer primary key autoincrement, _display_name text not null, _data text not null);”表示创建一个歌词文件的表,包含 _id, _display_name, _data 三列。其中 _id 是自增的且惟一的整数, _display_name 是歌词文件名称, _data 是歌词文件路径。歌词显示效果如图 3 所示。

4 系统测试

首先在 Android SDK 1.6 的模拟器上对影音播放器的各项功能进行调试和测试。调试结束后,将编译生成的安装文件(.APK)拷贝到 SD 卡,利用 Android 文件管理器将播放器软件安装到 OMAP3530 开发板的系统上,然后将测试用的视频文件、MP3 音乐文件及对应的歌词文件(.lrc)拷贝到 SD 卡,OMAP3530 开发板上的影音播放器可以通过文件浏览器选择音视频文件并

正常播放,在播放音频文件时可以正常显示匹配的歌词,支持快进、快退、暂停等控制功能。



图 3 歌词显示界面

5 结 语

本文以 Android 上影音播放器的开发为范例,详细给出了 Android 上应用的开发方法。开发的影音播放器在 Android 手机模拟器和 OMAP3530 开发板上测试通过。

参 考 文 献

- [1] 百度百科. Android[EB/OL]. [2009-07-23]. http://baike.baidu.com/view/1241829.htm?fr=ala0_1_1.
- [2] 叶达峰. Eclipse 编程技术与实例[M]. 北京:人民邮电出版社,2006.
- [3] 姚昱旻,刘卫国. Android 的架构与应用开发研究[J]. 计算机系统应用,2008(11):110-112.
- [4] 陈璟,陈平华,李文亮. Android 内核分析[J]. 现代计算机,2009(11):112-115.
- [5] 田森,李大和. 用 Android 开发手机应用[J]. 程序员,2008(1):56-61.
- [6] 赵亮,张维. 基于 Android 技术的界面设计与研究[J]. 电脑知识与技术,2009,29(5):8183-8185.
- [7] 公磊,周聪. 基于 Android 的移动终端应用程序开发与研究[J]. 计算机与现代化,2008(8):85-89.
- [8] Texas Instruments. OMAP3530/25 applications processor [EB/OL]. [2008-02-17]. <http://www.ti.com/cn/lit/pdf/sprs507>, 2008.
- [9] 佚名. MP3 标签格式[EB/OL]. [2008-03-16]. http://blog.sina.com.cn/s/blog_545a9a6d01009bvl.html.

作者简介:张正政 男,1987 年出生,湖北蕲春人,硕士研究生。主要研究方向为嵌入式系统。

林耀荣 男,1972 年出生,广东江门人,高级工程师。主要研究方向为嵌入式系统、数字图像处理。