

# 手机平台应用开发实验报告

学号： 1433198

班级： 周五下午班

姓名： 黄建武

实验名称： Lab6

## 一． 参考资料

<https://developer.android.com/guide/index.html> Android 官方文档

[http://blog.sina.com.cn/s/blog\\_5da93c8f0102vcam.html](http://blog.sina.com.cn/s/blog_5da93c8f0102vcam.html) Android

获取内外置存储卡方法

<http://www.jizhuomi.com/android/example/104.html> 安卓音乐播

放器开发实例

## 二． 实验环境

系统： Windows 10

工具： Android Studio 2.1.3

JDK： 1.8

minSdk： 19

targetSdk： 24

## 三． 实验目的

1. 学会使用 MediaPlayer
2. 学会简单的多线程编程，使用 Handle 更新 UI
3. 学会使用 Service 进行后台工作
4. 学会使用 Service 与 Activity 进行通信

## 四． 实验步骤

1. 阅读实验要求文档。
2. 运行 TA 的 Demo 程序，了解实验要求细节。
3. 首先完成页面的布局, 页面整体使用 `LinearLayout`, 控件按垂直布局, 中间的进度条和时间是 `RelativeLayout`, 下面的按钮是一个 `LinearLayout`, 控件按水平布局。因为按钮的样式都一样, 所以直接使用 `style`, 复用代码。

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_horizontal"
    android:orientation="vertical">

    <ImageView...>

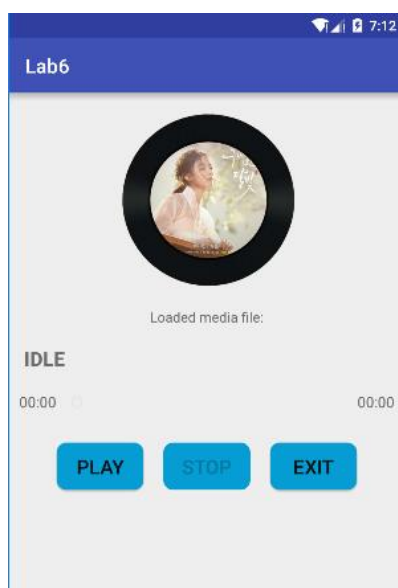
    <TextView...>
    <TextView...>

    <TextView...>

    <RelativeLayout...>

    <LinearLayout...>
</LinearLayout>
```

页面布局如下



4. 添加音乐服务的类，在 AndroidManifest.xml 文件里注册 Service，音乐播放器共有 4 种状态，IDLE, Playing, Pause, Stop，根据状态处理对应的按钮点击事件，在 onCreate 函数中载入音乐文件，初始化音乐播放器。

```
@Override
public void onCreate() {
    super.onCreate();
    mediaPlayer = new MediaPlayer();

    // 交作业使用内置的mp3
    // mediaPlayer = MediaPlayer.create(this, R.raw.music);
    // path = "Built-in";

    // 使用本地sd卡的mp3
    try {
        path = Environment.getExternalStorageDirectory().getAbsolutePath() + "/data/music.mp3";
        mediaPlayer.setDataSource(path);
        mediaPlayer.prepare();
    } catch (IOException e) {
        e.printStackTrace();
    }

    mediaPlayer.setLooping(true);
}
```

5. 实现按钮的点击事件，播放暂停停止退出的功能，MainActivity 实现 View.OnClickListener 的接口，不用每个按钮都写一个监听器。
6. 进行调试，完善逻辑上的错误和考虑不周全的地方，如 Stop 后再 play 程序崩溃，在停止前将音乐跳回文件开头，复位进度条，删除回调函数。

```
case R.id.stop:
    status.setText("Stop");
    play.setText("PLAY");
    stop.setEnabled(false);
    musicService.mediaPlayer.seekTo(0);
    musicService.stop();
    seekBar.setProgress(0);
    handler.removeCallbacks(seekBarRunnable);
    handler.removeCallbacks(rotation);
    break;
```

7. 添加多线程服务，使用 Handle 处理，实现进度条显示播放时间，拖动进度条等功能，主要重写了 SeekBar 的 onProgressChanged 函数，当进度条变化的时候更新播放时间，当拖动进度条时音乐跳到对应位置。

```
@Override
public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
    int playedtime = (int) (progress * 1.0 * musicService.mediaPlayer.getDuration() / seekBar.getMax());
    String current = time.format(playedtime);
    starttime.setText(current);
    if (fromUser)
        musicService.mediaPlayer.seekTo((int) (progress * 1.0 * musicService.mediaPlayer.getDuration() /
seekBar.getMax())); // 跳到该曲该秒
}
```

在线程中只需要改变进度条的进度即可，它会自动触发 onProgressChanged。

```
seekBarRunnable = new Runnable() {
    @Override
    public void run() {
        int process = (int) (musicService.mediaPlayer.getCurrentPosition() * 1.0 * seekBar.getMax() /
musicService.mediaPlayer.getDuration());
        seekBar.setProgress(process);
        handler.postDelayed(seekBarRunnable, 100);
    }
};
```

8. 实现图片旋转功能，暂停或停止播放时 removeCallBack

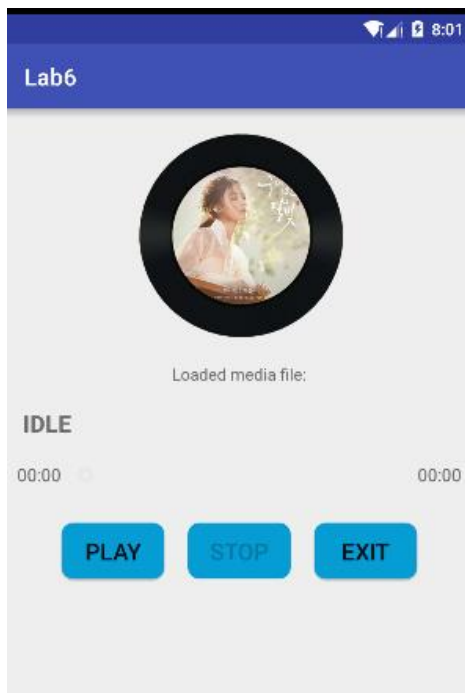
```
rotation = new Runnable() {
    @Override
    public void run() {
        cover.setRotation((cover.getRotation() + 1));
        handler.postDelayed(rotation, 100);
    }
};
```

9. 实现使用本地音乐文件，运行时在代码中选择不同的载入方式

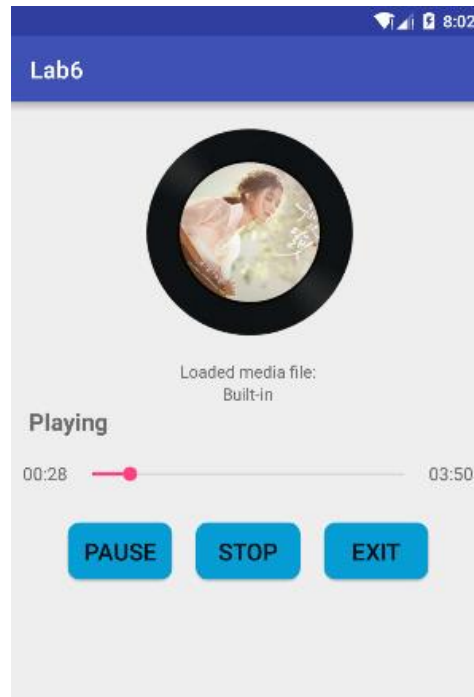
```
try {
    path = Environment.getExternalStorageDirectory().getAbsolutePath() + "/data/music.mp3";
    mediaPlayer.setDataSource(path);
    mediaPlayer.prepare();
} catch (IOException e) {
    e.printStackTrace();
}
```

## 五． 实验结果截图

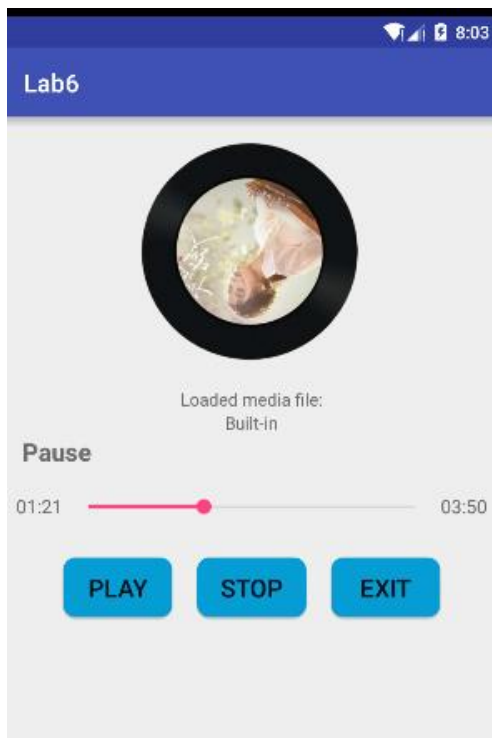
主页面



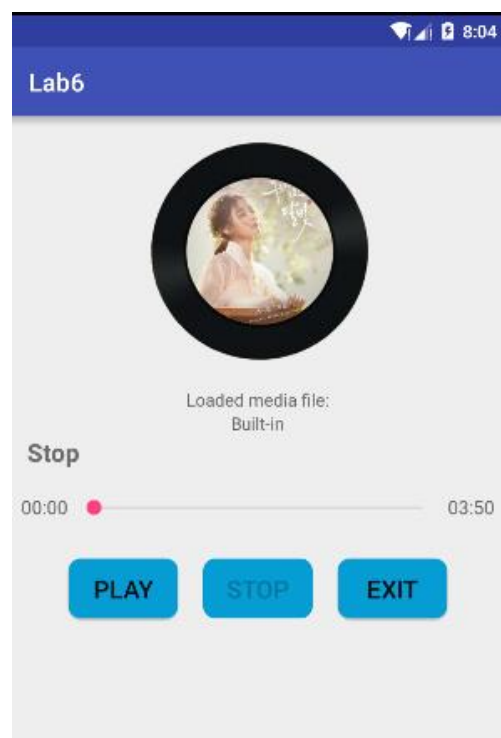
播放音乐



暂停



停止



## 六． 实验过程遇到的问题

1. 程序崩溃, 空指针异常

```
seekBar.setProgress(0);  
seekBar.setMax(musicService.mediaPlayer.getDuration());|
```

显示这行抛出异常，而 seekBar 肯定不会为空，那就是 mediaPlayer 为空，仔细检查代码后发现我是在第一次点击 PLAY 按钮时才载入音乐文件，在这之前都没有为 mediaPlayer 赋值，于是在布局文件中设置 SeekBar 的 max 值，然后将音乐总的毫秒数一一对应到进度条上。

## 2. 拖动进度条歌曲总是从头播放.

拖动进度条后我使用了 seekTo 让歌曲跳到对应的位置播放，从头播放那应该就是参数值为 0，原来犯了一个很低级的错误，  
`seekBar.getProgress()/seekBar.getMax()*musicService.mediaPlayer.getDuration()`，第一项整除永远等于 0，于是改成下面这样

```
(int) (seekBar.getProgress()*1.0*musicService.mediaPlayer.getDuration()/seekBar.getMax())
```

## 3. 在模拟器中拖动进度条很鬼畜，拖到某个位置放手进度条反弹，继续从刚才的位置播放，试了很久都不行，没有跳证明 seekTo 应该没执行，当用真机调试时却一点问题都没有，拖动进度条一切正常，输出 fromUser 才发现在模拟器上拖动进度条时 false，所以下面的 seekTo 没有被执行，可能是模拟器的

问题也可能是 API，手机的系统时 Android4.4。

```

    public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
        int playedtime = (int) (progress * 1.0 * musicService.mediaPlayer.getDuration() / seekBar.getMax());
        String current = time.format(playedtime);
        starttime.setText(current);
        if (fromUser) {
            musicService.mediaPlayer.seekTo((int) (seekBar.getProgress() * 1.0 * musicService.mediaPlayer
                .getDuration() / seekBar.getMax())); //跳到该曲该秒
        }
    }
}

```

#### 4. 按返回键回到桌面程序就崩溃了

按 Home 键返回桌面程序能正常在后台运行,应该是按了返回键之后 Activity 被结束了而音乐服务还没有释放资源所以导致程序崩溃,所以加上了对返回键的监听,当按下返回键的时候不要结束进程,转为后台运行.

```

@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if (keyCode == KeyEvent.KEYCODE_BACK) {
        moveTaskToBack(false);
        return true;
    }
    return super.onKeyDown(keyCode, event);
}

```

## 七. 思考与总结

通过本次实验,实现了一个简单的音乐播放器,学会使用多线程编程的基础,在线程里面更新 UI,使用 Service 进行后台工作。本次实验也遇到了一些问题,一开始写的时候没有想好逻辑,等待播放暂停停止这几个状态的转换没有仔细考虑清楚,写代码的时候会出现各种 bug,总有情况没考虑周全,于是仔细分析了各种状态的转换,在 MusicService 类中添加了几个状态变量,理清了思路后修改了代码,以后应该在实验前先理清思路才能节省时间事半功倍。