

# 中山大学移动信息工程学院本科生实验报告

## ( 2017 年秋季学期 )

课程名称：移动应用开发

任课教师：郑贵锋

年级	2015	专业 ( 方向 )	移动信息工程 ( 互联网方向 )
学号	15352151	姓名	柯博
电话	13671412922	Email	kebo@@mail2.sysu.edu.cn
开始日期		完成日期	

### 一、 实验题目

#### 实验六 服务与多线程——简单音乐播放器

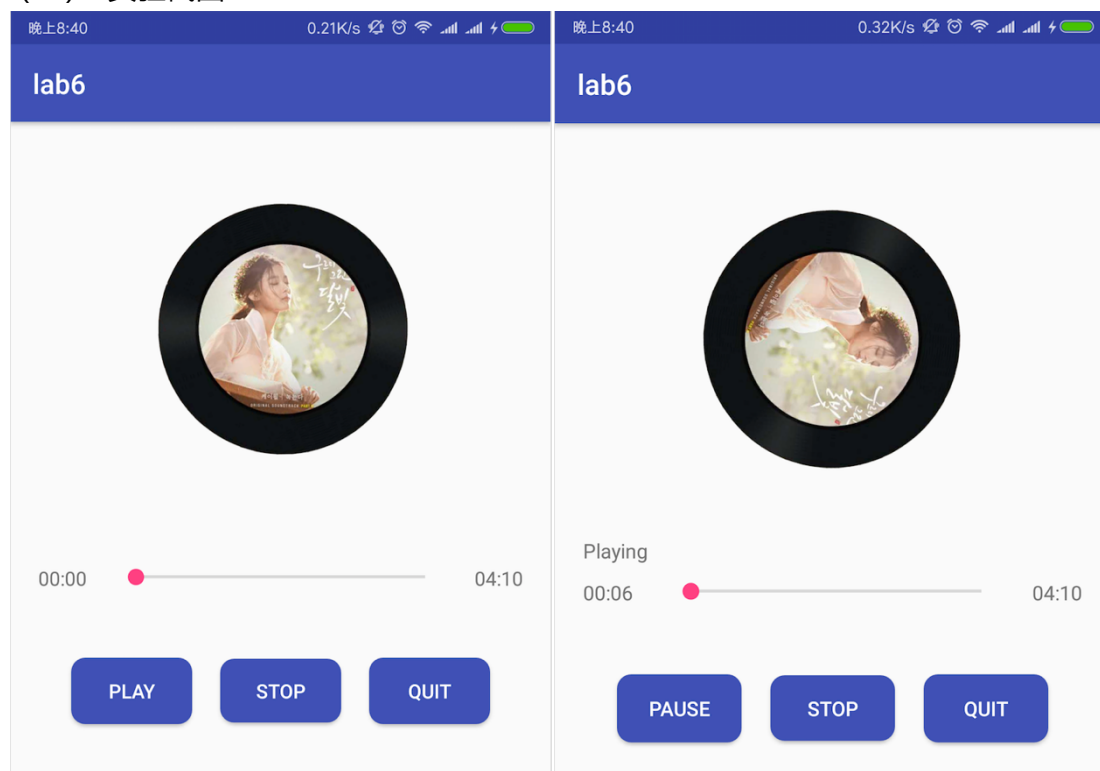
### 二、 实现内容

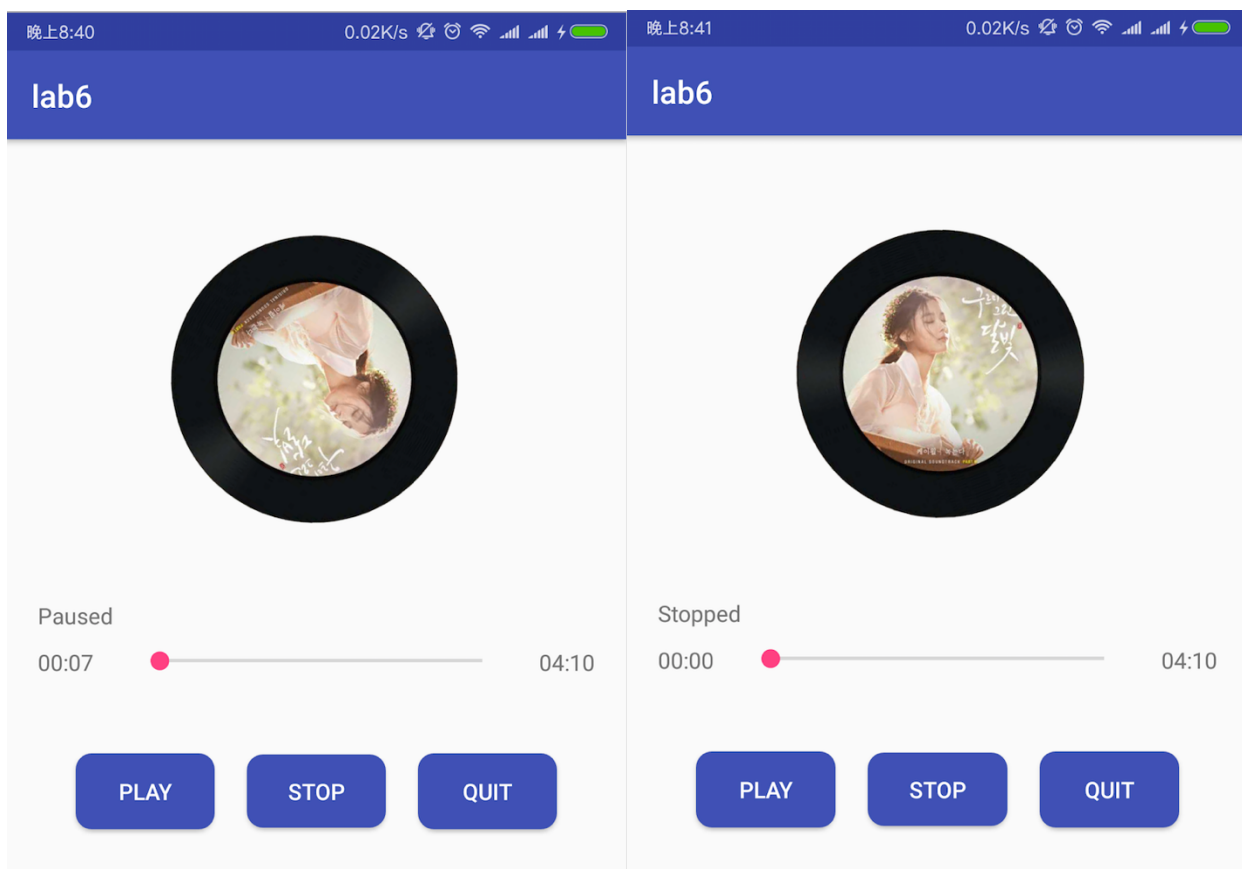
实现一个简单的音乐播放器，要求功能有：

1. 播放、暂停、停止、退出功能；
2. 后台播放功能；
3. 进度条显示播放进度、拖动进度条改变进度功能；
4. 播放时图片旋转，显示播放时间功能；

### 三、 课堂实验结果

#### ( 1 ) 实验截图





## (2) 实验步骤以及关键代码

设置三个按钮的点击监听事件，使用 `ibinder` 向 `Service` 发控制指令，调用 `refreshstatus` 函数：

```
MyPermission.verifyStoragePermissions(activity: this);

btn_play.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        try {
            Parcel reply = Parcel.obtain();
            ib_binder.transact( code: 1, data: null, reply, flags: 0);
            play_status = reply.readInt();
            refreshStatus();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
});

btn_stop.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        try {
            Parcel reply = Parcel.obtain();
            ib_binder.transact( code: 2, data: null, reply, flags: 0);
            play_status = reply.readInt();
            refreshStatus();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
});

btn_quit.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        onDestroy();
    }
});
```

refreshstatus 函数:

```
public void refreshStatus() {
    switch (play_status) {
        case 0:
            tv_status.setText("Stopped");
            btn_play.setText("PLAY");
            is_init = 1;
            oa_collection.end();
            break;
        case 1:
            tv_status.setText("Paused");
            btn_play.setText("PLAY");
            oa_collection.pause();
            is_init = 0;
            break;
        case 2:
            tv_status.setText("Playing");
            btn_play.setText("PAUSE");
            if (is_init == 1)
                oa_collection.start();
            else
                oa_collection.resume();
            break;
    }
}
```

与 Service 建立连接的 ServiceConnection 函数, 用 Intent 启动 Service:

```
serv_conn = new ServiceConnection() {
    @Override
    public void onServiceConnected(ComponentName name, IBinder service) {
        Log.d( tag: "service", msg: "connected");
        ib_binder = service;
        try {
            Parcel reply = Parcel.obtain();
            ib_binder.transact( code: 6, data: null, reply, flags: 0);
            int position = reply.readInt();
            skb_pos.setMax(position);
            tv_end.setText(sdf_pos.format(position));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    @Override
    public void onServiceDisconnected(ComponentName name) {
        serv_conn = null;
    }
};

Intent it_serv = new Intent( packageContext: this, MusicService.class);
startService(it_serv);
bindService(it_serv, serv_conn, Context.BIND_AUTO_CREATE);
```

进度条 SeekBar 的监听函数，其中 is\_changing 变量是用来标记是否对 SeekBar 有操作：

```
is_changing_pos = 0;
skb_pos.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
    @Override
    public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
        tv_now.setText(sdf_pos.format(progress));
    }

    @Override
    public void onStartTrackingTouch(SeekBar seekBar) { is_changing_pos = 1; }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {
        try {
            Parcel data = Parcel.obtain();
            data.writeInt(skb_pos.getProgress());
            ib_binder.transact( code: 5, data, reply: null, flags: 0);
        } catch (Exception e) {
            e.printStackTrace();
        }
        is_changing_pos = 0;
    }
});
```

与 Service 进行交互的 handler 函数，更新播放进度条：

```
final Handler hd_refresher = new Handler() {
    @Override
    public void handleMessage(Message msg) {
        super.handleMessage(msg);
        switch (msg.what) {
            case 1:
                if(is_changing_pos == 0) {
                    try {
                        Parcel reply = Parcel.obtain();
                        ib_binder.transact( code: 4, data: null, reply, flags: 0);
                        int position = reply.readInt();
                        skb_pos.setProgress(position);
                        if(position>0 &&!oa_collection.isRunning()) oa_collection.start();
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                }
                break;
        }
    }
};
```

新开一个线程，每 100 毫秒使用 handler 函数与 Service 通信，获得当前播放时间：

```
Thread t_refresh = new Thread() {
    @Override
    public void run() {
        while (true) {
            try {
                Thread.sleep( millis: 100);
            } catch (Exception e) {
                e.printStackTrace();
            }
            if (serv_conn != null) {
                hd_refresher.obtainMessage( what: 1).sendToTarget();
            }
        }
    }
};
t_refresh.start();

oa_collection = ObjectAnimator.ofFloat(iv_collection,
    propertyName: "rotation", ...values: 0, 359);
oa_collection.setDuration(20000);
oa_collection.setInterpolator(new LinearInterpolator());
oa_collection.setRepeatCount(ObjectAnimator.INFINITE);
oa_collection.setRepeatMode(ObjectAnimator.RESTART);
is_init = 1;
tv_status.setText("");
```

申请文件存储权限的 Permission 函数:

```
public class MyPermission {

    private static final int REQUEST_EXTERNAL_STORAGE = 1;
    private static String[] PERMISSIONS_STORAGE = {
        Manifest.permission.READ_EXTERNAL_STORAGE,
        Manifest.permission.WRITE_EXTERNAL_STORAGE
    };

    public static void verifyStoragePermissions(Activity activity) {
        int permission = ActivityCompat.checkSelfPermission(activity,
            permission: "android.permission.READ_EXTERNAL_STORAGE");

        if (permission != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(activity,
                PERMISSIONS_STORAGE, REQUEST_EXTERNAL_STORAGE);
        }
    }
}
```

MusicService 构造函数, 获取内置存储中的歌曲:

```
public MusicService() {
    try {
        mp.setDataSource(Environment.getExternalStorageDirectory()+"/Download/melt.mp3");
        mp.prepare();
        mp.setLooping(true);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

重载 IBinder 的 transact 函数, 各操作码对应播放器的操作:

```
public class MusicBinder extends Binder {
    @Override
    public boolean onTransact(int code, Parcel data, Parcel reply, int flags)
        throws RemoteException {
        switch (code) {
            case 1: // start/pause
                if (mp.isPlaying()) {
                    mp.pause();
                    reply.writeInt(1);
                } else {
                    mp.start();
                    reply.writeInt(2);
                }
                break;
            case 2: // stop
                if (mp != null) {
                    mp.stop();
                    reply.writeInt(0);
                    try {
                        mp.prepare();
                        mp.seekTo(msec: 0);
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                }
                break;
            case 3: // exit
                mp.release();
                mp = null;
                break;
            case 4: // refresh
                reply.writeInt(mp.getCurrentPosition());
                break;
            case 5: // move the bar
                mp.seekTo(data.readInt());
                break;
            case 6: // get max
                reply.writeInt(mp.getDuration());
                break;
        }
        return super.onTransact(code, data, reply, flags);
    }
}
```

### （3）实验遇到困难以及解决思路

首先，最大的问题是无法获取到 mp3 文件，后来发现是 Manifest 中的权限写错了；接着发现点击各按钮时播放器的响应不如预期，仔细检查交互函数和 Service 的 handle 代码逻辑，理清操作和反馈的关系之后，问题解决；最后发现六一个 bug：在设置让歌曲播放之后，点击手机返回键退出 activity，此时歌曲仍在播放，再次打开 app 发现图片动画不会动了。

## 四、 课后实验结果

经过思考，上文提到过的动画 bug 是因为代码中将动画的启动与按钮进行了绑定，MainActivity 的 onCreate 函数中只定义了动画的一些属性，但是没有动画 start。修改思路为在获取当前播放时间的 handler 函数（每秒都会与 Service 交互的）中增加检查当前播放状态，若歌曲处于播放中，则启动动画。

## 五、 实验思考及感想

无

作业要求：

1. 命名要求: 学号\_姓名\_实验编号，例如 15330000\_林 XX\_lab1。
2. 实验报告提交格式为 pdf。
3. 实验内容不允许抄袭，我们要进行代码相似度对比。如发现抄袭，按 0 分处理。