Service 生命周期实验

一．实验目的

了解 Android Service 的生命周期

二．实验条件

1. Windows、Ubuntu11.04 或其他兼容的 Linux 操作系统
2. JDK（建议安装 JDK8 及其以上版本）、Android Studio（建议安装 v0.4.2 版）

三．实验原理

1. Service 生命周期

在 Service 的生命周期中，被回调的方法比 Activity 少一些，只有 onCreate, onStartCommand, onDestroy, onBind 和 onUnbind。通常有两种方式启动一个 Service, 他们对 Service 生命周期的影响是不一样的。

① 通过 startService 启动

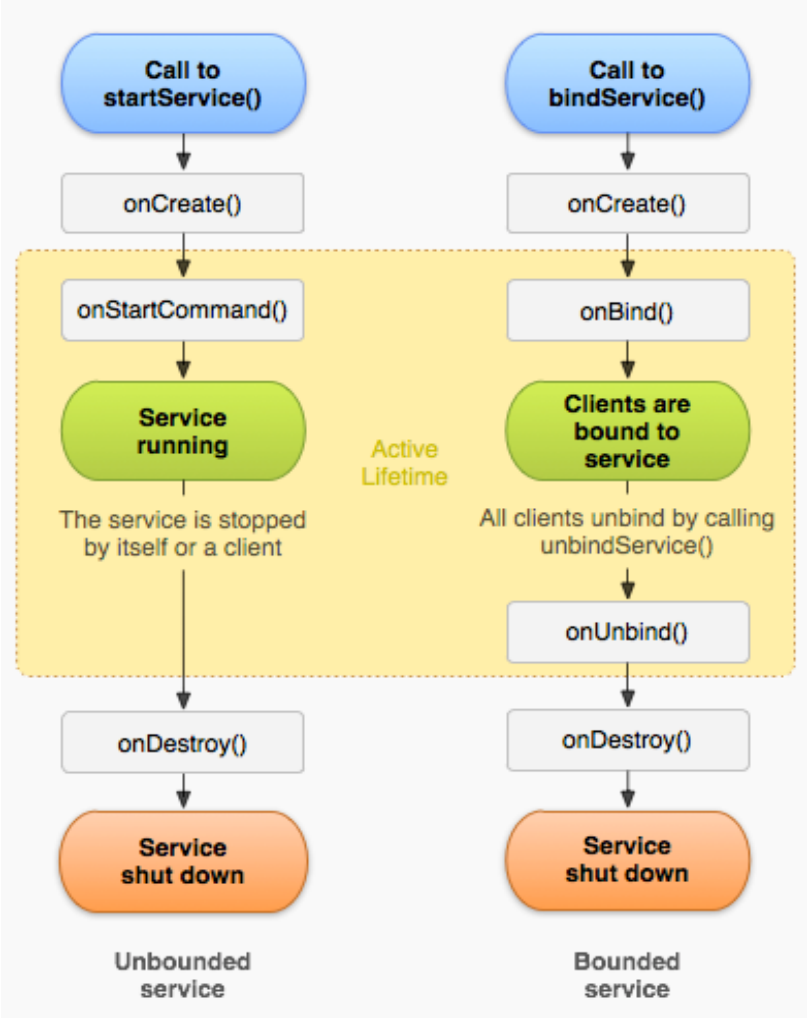
在同一个应用任何地方调用 startService() 方法就能启动 Service 了，然后系统会回调 Service 类的 onCreate() 以及 onStartCommand() 方法。这样启动的 Service 会一直运行在后台，直到 Context.stopService() 或者 selfStop() 方法被调用。另外如果一个 Service 已经被启动，其他代码再试图调用 startService() 方法，是不会执行 onCreate() 的，但会重新执行一次 onStartCommand ()。

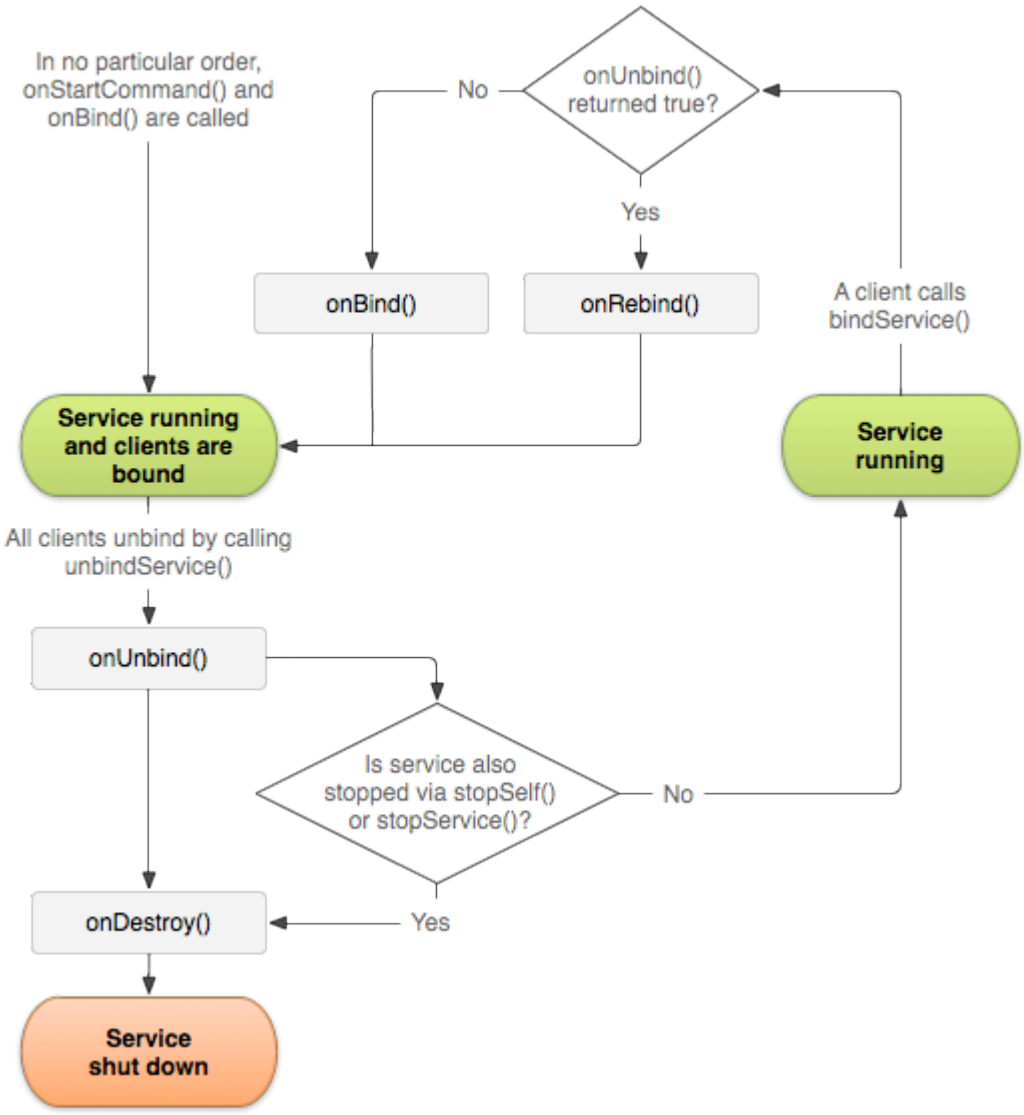
② 通过 bindService 启动

通过 bindService 是将这个 Service 和调用 Service 的客户类绑起来，如果调用这个客户类被销毁，Service 也会被销毁。用这个方法的一个好处是，bindService() 方法执行后 Service 会回调上边提到的 onBind() 方

法，你可以从这里返回一个实现了 IBind 接口的类，在客户端操作这个类就能和这个服务通信了，比如得到 Service 运行的状态或其他操作。如果 Service 还没有运行，使用这个方法启动 Service 就会调用 onCreate() 方法而不会调用 onStartCommand ()。

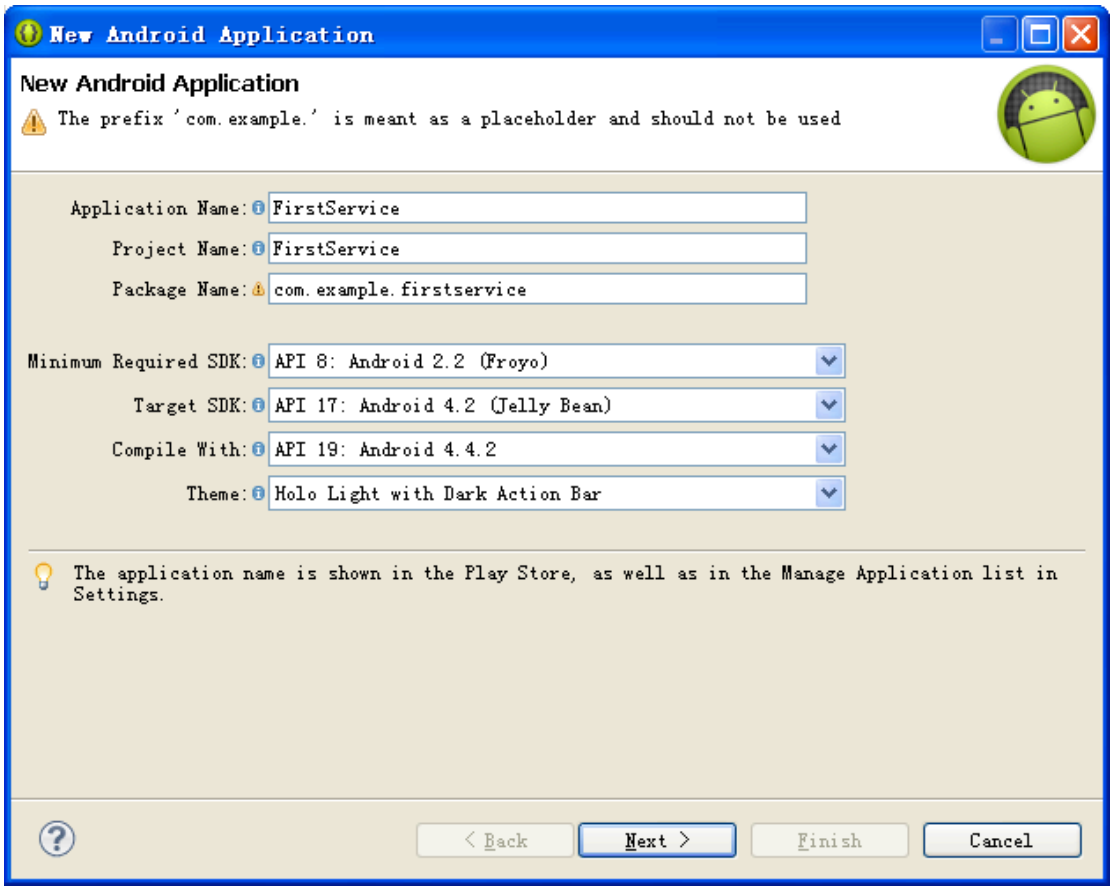
Service 生命周期如图所示：

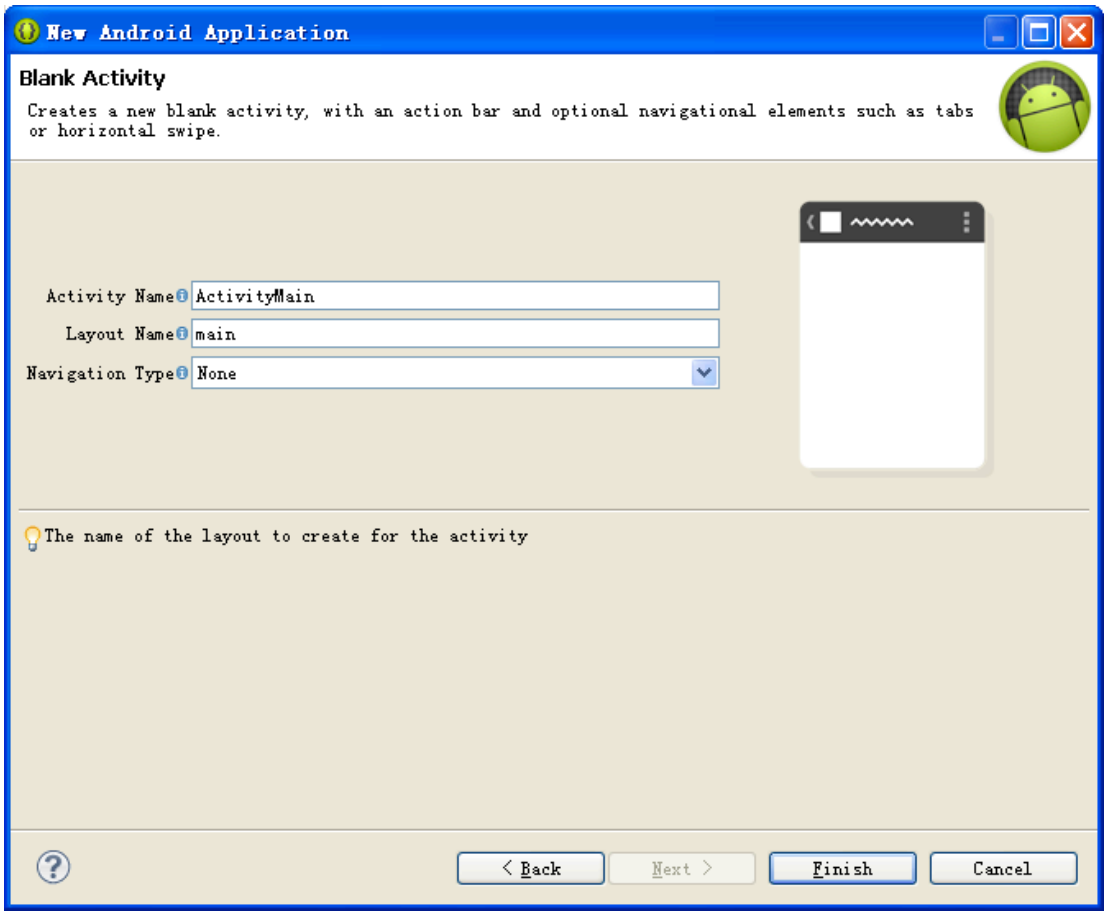




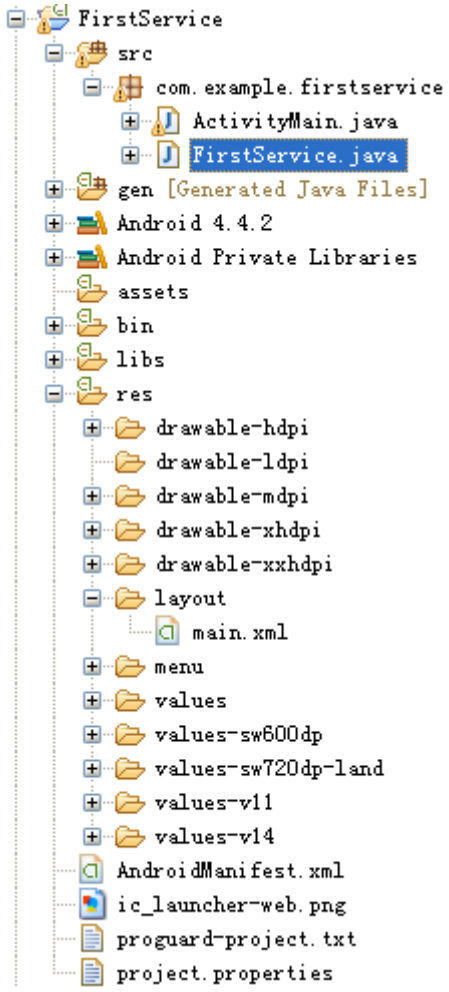
四．实验步骤

1. 新建一个 **Android** 工程，命名为 **FirstService**，如下图所示。


New Android Application
New Android Application
The prefix 'com.example.' is meant as a placeholder and should not be used
Application Name: FirstService
Project Name: FirstService
Package Name: com.example.firstservice
Minimum Required SDK: API 8: Android 2.2 (Froyo)
Target SDK: API 17: Android 4.2 (Jelly Bean)
Compile With: API 19: Android 4.4.2
Theme: Holo Light with Dark Action Bar
The application name is shown in the Play Store, as well as in the Manage Application list in Settings.
< Back
Next >
Finish
Cancel


New Android Application
Blank Activity
Creates a new blank activity, with an action bar and optional navigational elements such as tabs or horizontal swipe.
Activity Name: ActivityMain
Layout Name: main
Navigation Type: None
The name of the layout to create for the activity
< Back
Next >
Finish
Cancel

2. 建立一个 Service 类，命名为 FirstService.java



在 FirstService.java 中复写 Service 的 onCreate，onDestroy 等方法，并添加一些打印信息，添加打印信息的方式与之前 Activity 生命周期实验相同，这里不再累述，其代码内容如下：

```
package com.example.firstservice;

import android.app.Service;
import android.content.Intent;
import android.os.IBinder;
import android.os.Binder;
import android.util.Log;

public class FirstService extends Service{

    private static final String TAG = "FirstService";

    private MyBinder Binder = new MyBinder();
```

```java
@Override
public IBinder onBind(Intent arg0) {
    // TODO Auto-generated method stub
    Log.e(TAG, "start onBind!!");
    return Binder;
}

@Override
public void onCreate() {
    // TODO Auto-generated method stub
    Log.e(TAG, "start onCreate!!");
    super.onCreate();
}

@Override
public void onDestroy() {
    // TODO Auto-generated method stub
    Log.e(TAG, "start onDestroy!!");
    super.onDestroy();
}

@Override
public int onStartCommand(Intent intent, int flags, int startId) {
    // TODO Auto-generated method stub
    Log.e(TAG, "start onStartCommand!!");
    Log.e(TAG, "flags-->" + flags);
    Log.e(TAG, "startId-->" + startId);
    return super.onStartCommand(intent, flags, startId);
}

@Override
public boolean onUnbind(Intent intent) {
    // TODO Auto-generated method stub
    Log.e(TAG, "start onUnbind!!");
    return super.onUnbind(intent);
}


public class MyBinder extends Binder{
    public String getData(){
        return "test data!!";
    }
}
```

```
}
```

3.在 res/layout 文件中的布局文件 main.xml 中添加四个按钮，分别控制 Service 的启动，停止，Service 与 Activity 的绑定和解除绑定，其代码如下：

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".ActivityMain" >

    <TextView
        android:id="@+id/tv"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />

    <Button
        android:id="@+id/startservice"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/tv"
        android:text="startService" />

    <Button
        android:id="@+id/stopservice"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/startservice"
        android:text="stopService" />

    <Button
        android:id="@+id/bindservice"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/stopservice"
        android:text="bindService"/>

    <Button
```

```
        android:id="@+id/unbindservice"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/bindservice"
        android:text="unBindService"/>

</RelativeLayout>
```

类 **ActivityMain.java** 中的代码如下，这里主要通过四个按钮来实现 **Service** 的启动、停止，**Service** 与 **Activity** 的绑定和解绑：

```
package com.example.firstservice;

import com.example.firstservice.FirstService.MyBinder;

import android.os.Bundle;
import android.os.IBinder;
import android.app.Activity;
import android.content.ComponentName;
import android.content.Intent;
import android.content.ServiceConnection;
import android.util.Log;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class ActivityMain extends Activity {

    private static final String TAG = "ActivityMain";

    Button button1 = null;
    Button button2 = null;
    Button button3 = null;
    Button button4 = null;

    OnClickListener listener1 = null;
    OnClickListener listener2 = null;
    OnClickListener listener3 = null;
    OnClickListener listener4 = null;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        listener1 = new OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                Intent intent1 = new Intent(ActivityMain.this,
FirstService.class);
                startService(intent1);
            }
        };

        listener2 = new OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                Intent intent2 = new Intent(ActivityMain.this,
FirstService.class);
                stopService(intent2);
            }
        };

        listener3 = new OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                Intent intent3 = new Intent(ActivityMain.this,
FirstService.class);
                bindService(intent3, conn, BIND_AUTO_CREATE);
            }
        };

        listener4 = new OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                unbindService(conn);
            }
        };
```

```
        button1 = (Button)findViewById(R.id.startservice);
        button1.setOnClickListener(listener1);
        button2 = (Button)findViewById(R.id.stopservice);
        button2.setOnClickListener(listener2);
        button3 = (Button)findViewById(R.id.bindservice);
        button3.setOnClickListener(listener3);
        button4 = (Button)findViewById(R.id.unbindservice);
        button4.setOnClickListener(listener4);
    }

    ServiceConnection conn = new ServiceConnection() {

        @Override
        public void onServiceDisconnected(ComponentName name) {
            // TODO Auto-generated method stub
        }

        @Override
        public void onServiceConnected(ComponentName name, IBinder
service) {
            // TODO Auto-generated method stub
            MyBinder binder = (MyBinder)service;
            String data = binder.getData();
            Log.e(TAG, data);
        }
    };


    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is
present.
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }

}
```

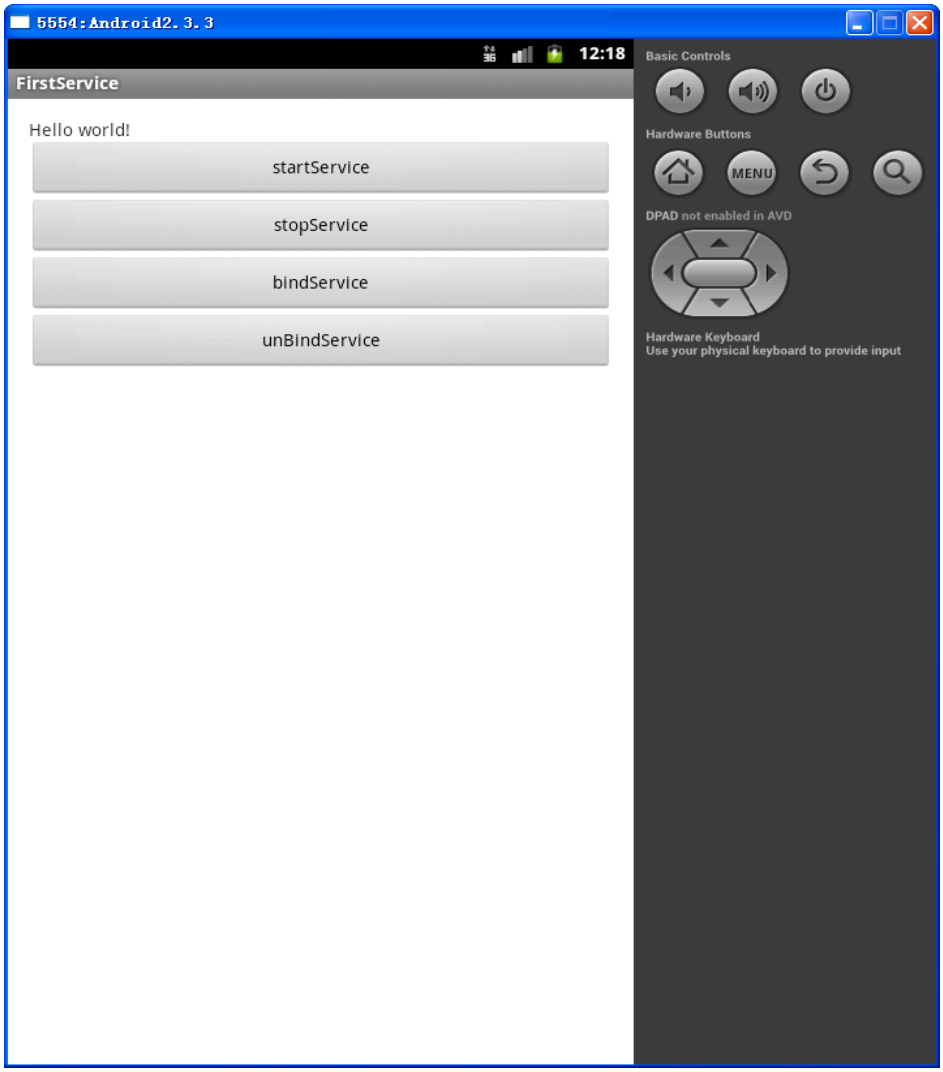
AndroidMainfest.xml 文件里添加 Service 声明，其代码如下：

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.firstservice"
```

```
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.firstservice.ActivityMain"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <service
            android:name=".FirstService">
        </service>

    </application>

</manifest>
```

五．实验结果演示

将代码按上述步骤书写完毕后，即可打开安卓虚拟机，运行代码，结果如下：



1. 点击 startService 按钮，在日志输出窗口显示如下：

L...	Time	PID	TID	Application	Tag	Text
E	03-19 12:34:56.019	334	334	com.example.firsts...	FirstService	start onCreate!!
E	03-19 12:34:56.049	334	334	com.example.firsts...	FirstService	start onStartCommand!!
E	03-19 12:34:56.049	334	334	com.example.firsts...	FirstService	flags-->2
E	03-19 12:34:56.049	334	334	com.example.firsts...	FirstService	startId-->1

2. 再次点击 startService 按钮，在日志输出窗口显示如下：

```
E  03-19 12:40:29.158  334  334  com.example.firsts...  FirstService  start onStartCommand!!
E  03-19 12:40:29.158  334  334  com.example.firsts...  FirstService  flags-->2
E  03-19 12:40:29.170  334  334  com.example.firsts...  FirstService  startId-->2
```

3. 点击 stopService 按钮，在日志输出窗口显示如下：

```
E  03-19 12:42:16.878  334  334  com.example.firsts...  FirstService  start onDestroy!!
```

4. 点击 bindService 按钮，在日志输出窗口显示如下：

```
03-19 12:43:49.058  334  334  com.example.firsts...  FirstService  start onCreate!!
03-19 12:43:49.078  334  334  com.example.firsts...  FirstService  start onBind!!
03-19 12:43:49.178  334  334  com.example.firsts...  ActivityMain  test data!!
```

5. 点击 unBindService 按钮，在日志输出窗口显示如下：

```
03-19 12:44:19.999    334    334    com.example.firsts...  FirstService    start onUnbind!!
03-19 12:44:19.999    334    334    com.example.firsts...  FirstService    start onDestroy!!
```

六．参考资料

https://developer.android.com/guide/components/services.html