

Activity 的启动模式

知识解析

应用（Application）

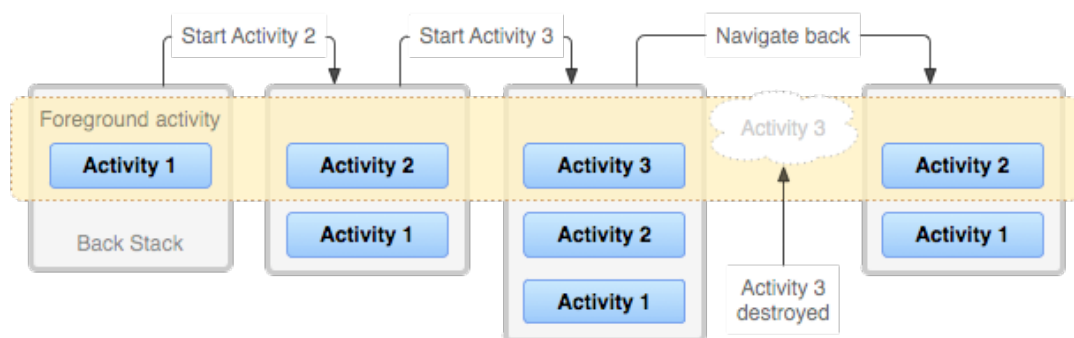
一个 apk 就是一个 Application，包含若干 activity/service/content provider/broadcast receiver 等以及其他资源

进程（Process）

一般来说，一个 Application 启动后，就是一个 process，linux 会给它分配一个 user id。这个应用的所有组件运行在这个进程中。默认的进程名称和该应用的包名相同，也可以通过 application 中的 android:sharedUserId 指定跟其他进程一样的 user id，此时这 2 个应用运行在同一个进程中。

Activity 堆栈（Activity Stack）

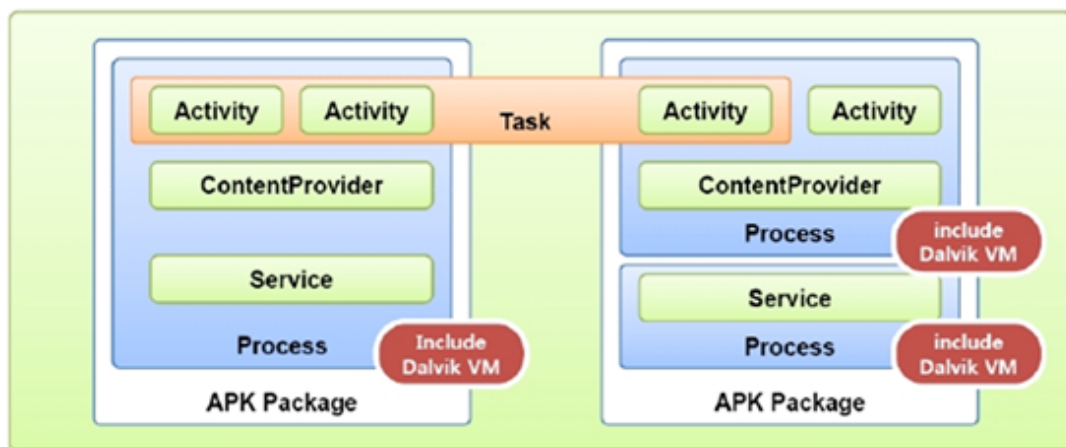
因为我们即使在一个应用中也不止会用到一个 activity，而是会用到多个 activity，为了避免多个 activity 的跳转发生混乱，android 使用 activity stack 的机制来管理这些 activity，采用先进后出的方式来控制它们。



任务（Task）

将多个 Activity 组合起来完成一个任务，这些 activity 的组合叫一个 Task。一个 Task 中的 activity 可能来自于同一个 App，也可能来自多个 App，例如：在使用微博的过程中，

如果要发一张图片，可以打开照相机来拍照，这时一般会启动系统的照相 App 来拍照。对于用户来说，他只是完成了一次正常的使用微博的功能。这些 activity 就组成了一个拍照并分享到微博的 task。



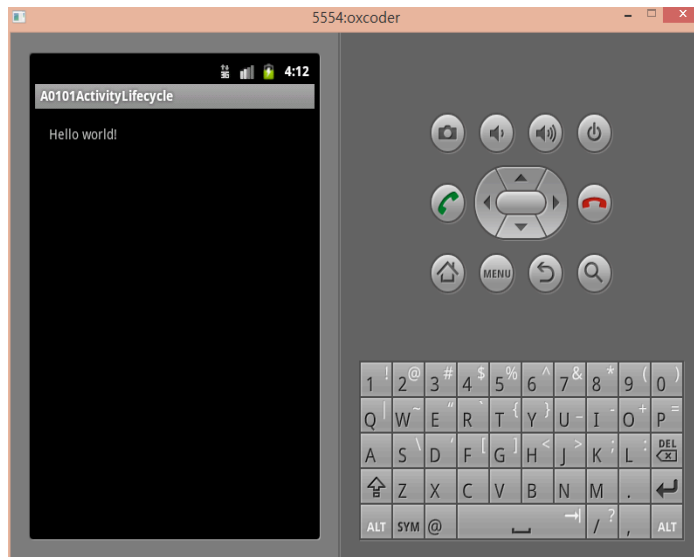
Activity 启动模式

在 android 里，有 4 种 activity 的启动模式，分别为：

- standard：每次启动都会新创建一个 activity 实例
- singleTop：如果在同一个任务(task)的 activity 栈顶已经有一个要启动的 activity 了，那么，这个 activity 将会接收对应的 intent（通过 onNewIntent()方法），而不会新创建 activity 实例。否则将会创建新的实例
- singleTask：和 singleInstance 类似，会启动一个新的 task 并且放在 activity stack 的底部（root）。如果已经有一个实例，则会通过调用 onNewIntent()方法接收这个 intent 而不会创建新的。可以允许其他的以 standard 或者 singleTop 方式启动的 activity 成为它所在任务的一部分
- singleInstance：不允许其他 activity 成为它所在 task 的一部分。这个 activity 是它所在任务唯一的成员。

一般不使用后面 2 种方式的启动方式

功能演示



操作实践

```
public class MainActivity extends Activity {  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        Log.e("Main", "onCreate");  
        setContentView(R.layout.activity_main);  
    }  
    protected void onPause() {  
        super.onPause();  
        Log.e("Main", "onPause");  
    }  
    protected void onResume() {  
        super.onResume();  
        Log.e("Main", "onResume");  
    }  
    protected void onStart() {  
        super.onStart();  
        Log.e("Main", "onStart");  
    }  
}
```

```
}
```

```
protected void onRestart() {  
    super.onRestart();  
    Log.e("Main", "onRestart");  
}  
protected void onStop() {  
    super.onStop();  
    Log.e("Main", "onStop");  
}  
protected void onDestroy() {  
    super.onDestroy();  
    Log.e("Main", "onDestroy");  
}  
}
```

职业素质

standard 每次都会新建，每个Task都可以有，且每个Task都可以有多个实例

singleTop 当前实例如果在栈顶，就不新建实例，调用其OnNewIntent。如不在栈顶，则新建实例

singleTask 新建一个Task，如果已经有其他的Task并且包含该实例，那就直接调用那个Task的实例

singleInstance 新建一个Task，且在该Task中只有它的唯一一个实例

在工作中我们根据相应的情况选择相应的启动模式，Activity作为Android最为基础的组件，它的启动模式和生命周期都是比较重要的。正确的选择它的启动模式能很好的优化我们的项目。