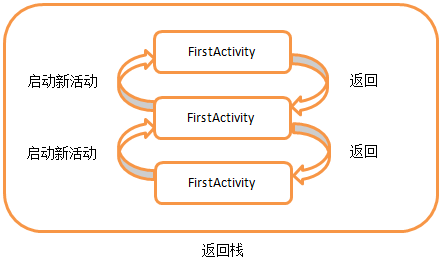
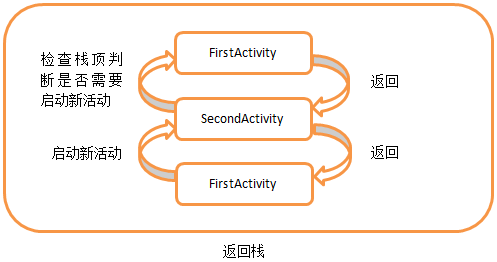
**Android启动模式**

参考：<http://blog.csdn.net/zhangjg_blog/article/details/10923643>

1. standard：默认的启动模式，每次启动都会创建该活动的一个新的实例。

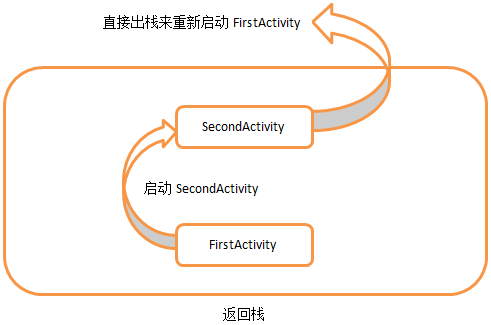


1. singleTop：



* 1. 启动活动时，若发现返回栈的栈顶已经是要启动的活动,则直接使用它,不会再创建新的活动实例，并调用该实例的onNewIntent()方法将Intent对象传递到这个实例中。
  2. 若要启动的活动已经存在于返回栈中，但是不在栈顶，依然会创建一个实例。

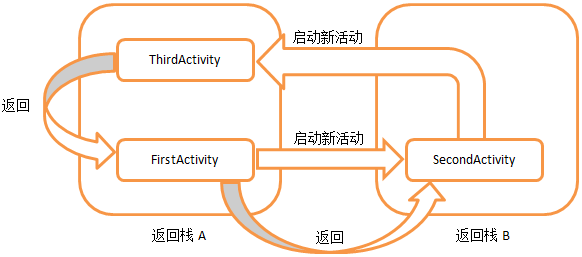
1. singleTask：在同一个任务中，只会有一个该activity的实例



要启动的activity设置为SingleTask启动模式时，将该启动模式为SingleTask的启动标志为FLAG\_ACTIVITY\_NEW\_TASK 且framework会检索是否已存在了一个taskAffinity属性值与要启动的activity的taskAffinity属性值相等的任务。

* 1. 若taskAffinity属性指定的任务存在，检查该任务中是否存在要启动Activity的实例
     + - * 若要启动的Activity实例存在，将该任务调到前台，清除要启动的Activity上面的所有Activity，显示要启动的Activity，并调用其onNewIntent()方法；
         * 若要启动的Activity实例不存在，在该任务中创建要启动的Activity的实例。
  2. 若taskAffinity属性指定的任务不存在，会创建一个新的由taskAffinity属性指定的任务，在该任务中创建要启动的Activity的实例。

1. singleInstance：整个系统中只会存在一个实例，且其独自占用一个任务



注：

1. 当在 ThirdActivity 的界面按下 Back 键,ThirdActivity 会从返回栈中出栈, FirstActivity 就成为了栈顶活动显示在界面上。在 FirstActivity 界面再次按下 Back 键,这时当前的返回栈已经空了,于是就显示了另一个返回栈的栈顶活动,即 SecondActivity。最后再次按下 Back 键,这时所有返回栈都已经空了,退出了程序。 如果要从ThirdActivity返回到SecondActivity，可以再次指定startActivity。
2. 被singleInstance模式的Activity开启的activity（被启动的activity，不是启动模式为singleInstance的activity），启动时系统也会为它加上FLAG\_ACTIVITY\_NEW\_TASK标志。
3. 要启动的activity是以FLAG\_ACTIVITY\_NEW\_TASK标志启动（与SingleTask启动模式相同），此时framework会检索是否已经存在了一个taskAffinity属性指定的任务
4. 若taskAffinity属性指定的任务存在，检查该任务中是否存在要启动Activity的实例
   * + - * 若要启动的Activity实例存在，将该任务调到前台，清除要启动的Activity上面的所有Activity，显示要启动的Activity，并调用其onNewIntent()方法；
         * 若要启动的Activity实例不存在，在该任务中创建要启动的Activity的实例。
5. 若taskAffinity属性指定的任务不存在，会创建一个新的由taskAffinity属性指定的任务，在该任务中创建要启动的Activity的实例。
6. manifest文件中<activity>的taskAffinity属性
7. taskAffinity表示一个任务，即当前activity所在的任务；
8. 具有相同的affinity的activity属于同一个任务；
9. 一个任务的affinity决定于这个任务的根activity（root activity）的taskAffinity；
10. 默认情况下，一个应用中的所有activity具有相同的taskAffinity，即应用程序的包名。当activity的taskAffinity设置为空字符串，表明这个activity不属于任何task。
11. 把不同的应用中的两个activity的taskAffinity设置成相同的值，这两个activity虽不在同一应用中，却会在运行时分配到同一任务中。
12. 如果要启动的activity为singleTask或singleInstance，启动该activity的activity中 onActivityResult收到RESULT\_CANCELED，直接取消了。
13. 当启动singleTask模式的activity时，该任务栈中只有一个实例。当按下返回键使回到上一个Activity时，当前singleTask模式的activity被销毁了，因此上一Activity无法拿到当前singleTask模式的activity返回的数据；
14. 当被启动和启动的activity为singleInstance模式时，两个activity不再同一个任务栈中，因此无法传值。

参考：<http://blog.csdn.net/androidhuoniao/article/details/6535042>