Activity的五种启动模式

江江安卓 郭霖 2022-07-28 08:00 发表于江苏

/ 今日科技快讯 /

微软和谷歌在美股盘后分别公布了季度财报，微软料本财年利润与销售均录得两位数增长、谷歌营收符合预期，广告客户需求强劲。两家公司盘后均涨超5%，带动纳指期货上涨1.5%。

/ 作者简介 /

本篇文章来自Zhujiang的投稿，文章主要分享了Android启动模式相关的内容，相信会对大家有所帮助！同时也感谢作者贡献的精彩文章。

Zhujiang的博客地址：

> https://blog.csdn.net/haojiagou/

/ 前因后果 /

这两天遇到了一个 bug ，说是应用打开一个二级页面，然后直接回到桌面，并不是杀掉应用，只是回到桌面，再次打开的时候没有回到那个二级页面，而是回到了首页。

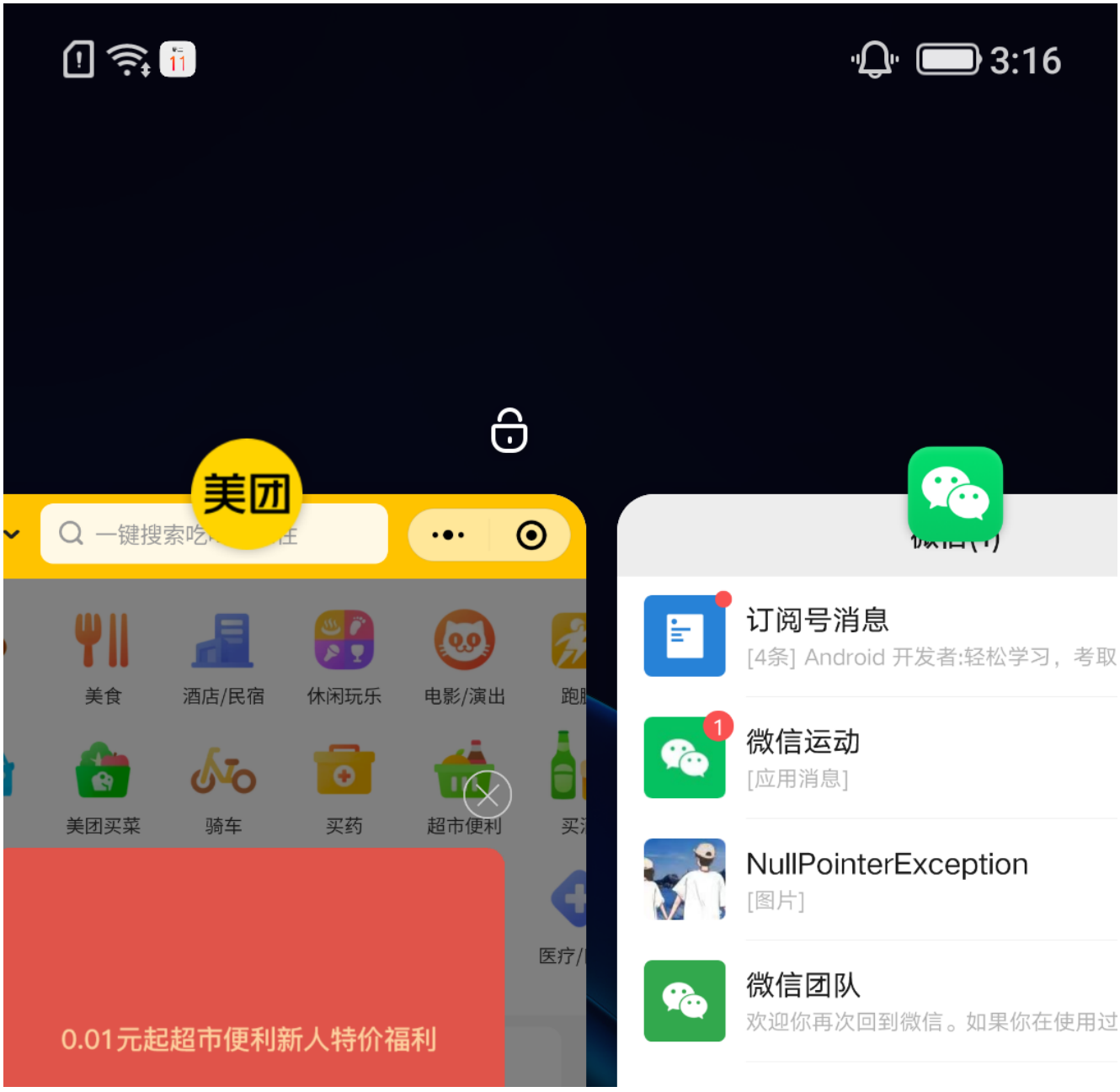
看到这里，很多人大概都知道是什么原因了，没错，就是 Activity 的启动模式设置为了 singleTask 而导致的问题，本来觉得自己基础还凑合，但这个问题真没有想到过，所以，今天来重新过一遍 Activity 的启动模式。

/ 开整 /

首先大家都知道 Android 中使用任务栈来存储创建的 Activity ，栈是先进后出，这里的任务栈当然也一样，最先进入的页面在栈的最底部，当按返回键的时候，每按一次，一个 Activity 出栈，直到栈空为止，当栈中没有 Activity 时，系统就会回收此任务栈。

任务栈

那么问题来了，什么是任务栈呢？一个应用默认只有一个任务栈，当然也可以有多个任务栈，某信中就有。查看任务栈其实很简单，在 Android 设备中进入多任务就能看到，来看下某信的截图：



上图中的某信就有多个任务栈。

那么为啥会有启动模式一说呢？？？因为如果多次启动同一个 Activity 的时候，不管任务栈中是否存在这个 Activity，都会创建多个 Activity，而且还浪费了内存空间，所以 Google

为 Activity 的创建提供了几种启动模式。

之前的版本中是四种启动模式，但在 Android12（ S 31）之后的版本则修改为了五种启动模式，下面咱们来一个一个看。

准备工作

先来做下准备工作，创建一个空的应用，然后再创建两个 Activity 并打印它们的生命周期：

```
override fun onCreate(savedInstanceState: Bundle?) {
   super.onCreate(savedInstanceState)
   setContentView(R.layout.activity_two)
   Log.e(TAG, "onCreate: ")
}

override fun onStart() {
   super.onStart()
   Log.e(TAG, "onStart: ")
}

override fun onResume() {
   super.onResume()
   Log.e(TAG, "onResume: ")
}

override fun onPause() {
   super.onPause()
   Log.e(TAG, "onPause: ")
}

override fun onStop() {
   super.onStop()
   Log.e(TAG, "onStop: ")
}

override fun onDestroy() {
   super.onDestroy()
   Log.e(TAG, "onDestroy: ")
}
```

标准模式——standard

这个启动模式是最常见的，Activity 默认就是此启动模式。每启动一次 Activity，就会创建一个新 Activity 实例并置于栈顶。谁启动了这个 Activity，那么这个 Activity 就运行在启动它的那个 Activity 所在的栈中。

其实后面这句话挺重要，之前学习的时候并不是太理解这句话，但也是在不久前遇到了一个问题让我重新理解了：Android 中有多窗口模式，这个问题是在多窗口模式下发现的，我们应用中有个地方会调用设置中的页面来选择声音，在正常模式下是没有问题的，但是多窗口模式下就会重新启动一个任务栈，但我们系统中限制多窗口模式下只能有一个应用在前台，结果我们自己的应用被干掉了。。。大家一定引以为戒，知识点的每一句话都有可能有用！

下面咱们来测试下标准模式，先一步一步来，先从第一个页面跳转到第二个页面，看下 log：

```
E/MainActivity: onCreate:
E/MainActivity: onStart:
E/MainActivity: onResume:
E/MainActivity: onPause:
E/TwoActivity: onCreate:
E/TwoActivity: onStart:
E/TwoActivity: onResume:
E/MainActivity: onStop:
```

没什么问题，和预想的一致，然后回到桌面再打开应用，看下 log：

```
E/TwoActivity: onPause:
E/TwoActivity: onStop:
E/TwoActivity: onStart:
E/TwoActivity: onResume:
```

嗯，没问题，现在任务栈里有两个 Activity，点击返回键依次退出再来看下 log：

```
E/TwoActivity: onPause:
E/MainActivity: onStart:
E/MainActivity: onResume:
E/TwoActivity: onStop:
E/TwoActivity: onDestroy:
E/MainActivity: onPause:
E/MainActivity: onStop:
```

从第二个 Activity 回到第一个 Activity 可以理解，但是大家有没有发现第一个 Activity 并没有走 onDestroy ，这里引用下一个厉害的大哥文章中的描述吧：

Android 12 以前，当我们处于 Root Activity 时，点击返回键时，应用返回桌面，Activity 执行 onDestroy，程序结束。Android 12 起同样场景下 Activity 只会 onStop，不再执行 onDestroy。

到这里标准模式就差不多了，因为这是默认的启动模式，大家使用也最频繁，也就不再啰嗦。

栈顶模式——singleTop

栈顶模式其实很好理解，如果栈顶存在该activity的实例，则复用，不存在新建放入栈顶，它的表现几乎和 上面刚说的标准模式一模一样，栈顶模式的 Activity 实例可以无限多，唯一的区别是如果在栈顶已经有一个相同类型的 Activity 实例，那么 Intent 则不会再去创建一个 Activity，而是通过 onNewIntent() 发送到现有的Activity。

比如应用现在在一个详情页面，而且这个页面启动模式为栈顶模式，这个时候来了一个通知，点击通知正好要跳转到详情页面，那么这个时候任务栈就不会为这个 Activity 再创建一个实例而用已经在栈顶的之前创建好的 Activity 实例。

栈内复用——singleTask

这个模式之前真的没有理解透彻，之前我理解的就是如果栈内存在该 Activity 的实例则进行复用，如果不存在则创建。

接下来将刚才的 Demo 中的主 Activity 的启动模式改为栈内复用，先来看下启动应用后点击跳转到第二个 Activity 的 log：

```
E/MainActivity: onCreate:
E/MainActivity: onStart:
E/MainActivity: onResume:
E/MainActivity: onPause:
E/TwoActivity: onCreate:
E/TwoActivity: onStart:
```

```
E/TwoActivity: onResume:
E/MainActivity: onStop:
```

目前来看还是比较正常的，接下来直接回到桌面，再来看下 log：

```
E/TwoActivity: onPause:
E/TwoActivity: onStop:
```

也还对着呢，然后再次打开应用再看 log：

```
E/TwoActivity: onDestroy:
E/MainActivity: onStart:
E/MainActivity: onResume:
```

是不是不对了，我本来想让应用回到第二个 Activity，但为什么第二个 Activity 直接销毁了？

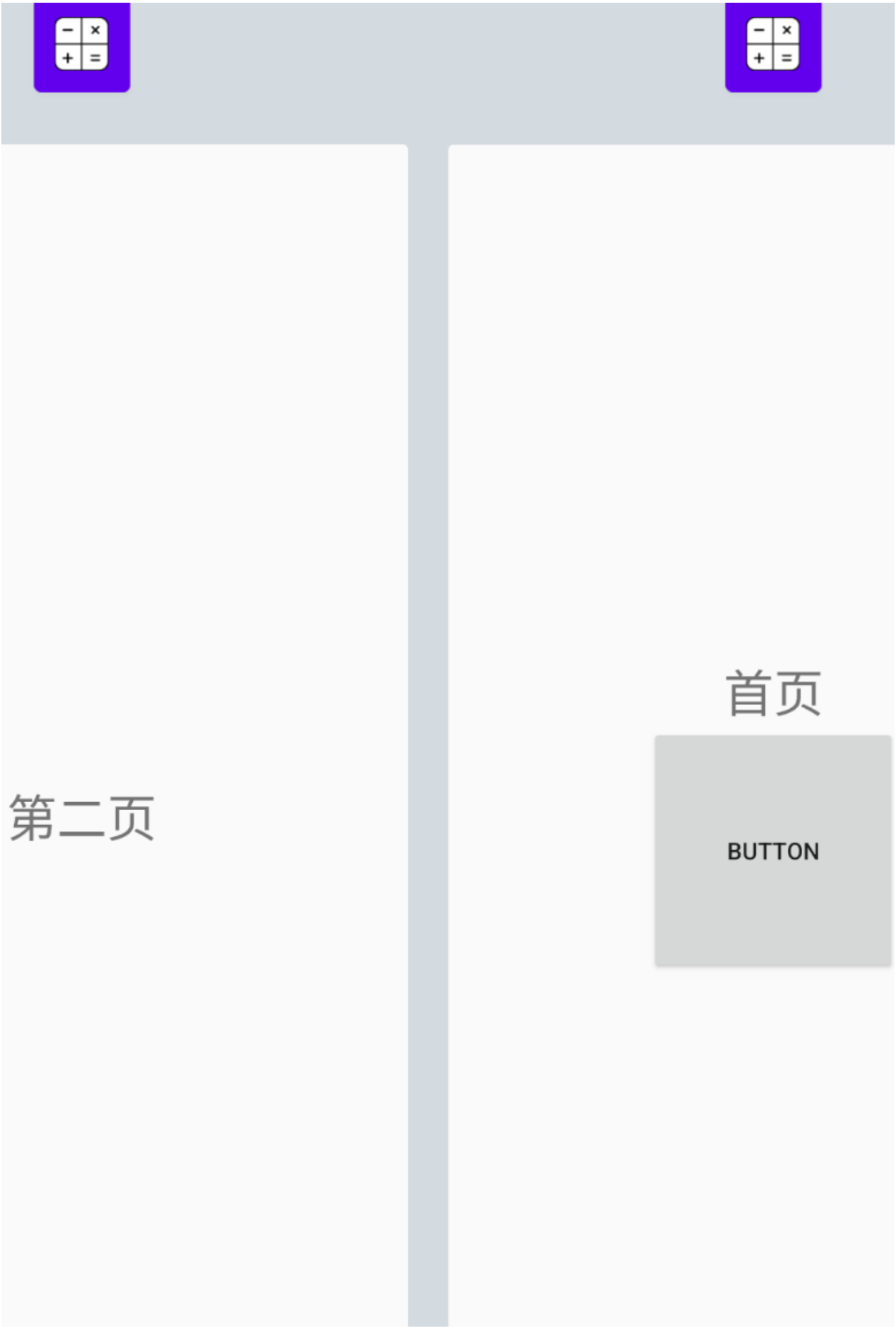
其实栈内复用中还有一点要注意，也正是我忽略的重要一点：栈内复用模式会将该实例上边的 Activity 全部出栈，将该实例置于栈顶，这也就是出现文章开头我说的那个问题的根本原因。

单例模式——singleInstance

单例模式，顾名思义，就是新开一个任务栈，该栈内只存放当前实例。比如说项目中语音通话功能，来电显示页面采用的就可以采用单例模式进行处理。

当然还有别的方法来新开任务栈，比如说启动 Activity 的时候加上 FLAG_ACTIVITY_NEW_TASK ，也会开启一个新的任务栈。

这里需要注意，即使将 Activity 的启动模式设置为单例模式或者添加了 flag，也不会出现像上面某信那种效果，因为 Activity 的 taskAffinity 是一样的，但如果将 Activity 的 taskAffinity 修改下，就可以出现类似于上面某信的效果，如下图所示：


首页
BUTTON
第二页

单例任务模式——singleInstancePerTask

其实这个单例任务模式是我自己编的😂，这个和上面所说的单例模式基本一致，只不过会为启动的 Activity 新建任务栈，而不需要像上面说的单例模式那样修改 taskAffinity。

/ 打完收工 /

到这里为止基本把 Android 的启动模式都说了一遍，为自己做个总结，以后不能再出现这种错误了。

推荐阅读：

我的新书，《第一行代码 第3版》已出版！

Android 13 Developer Preview一览

模仿Android微信小程序，实现小程序独立任务视图的效果

欢迎关注我的公众号

学习技术或投稿

▼

长按上图，识别图中二维码即可关注

阅读原文

喜欢此内容的人还喜欢

科技与狠活？JDK19中的虚拟线程到底什么鬼？

Hollis

太强大了！基于 Spring Cloud Alibaba微服务的架构系统，还支持分库分表、多租户等

编码超人

SpringBoot接入轻量级分布式日志框架GrayLog

码猿技术专栏