# 前言

我们要说点什么：

1.task是什么

巩固一下基础，即是：它可以帮助你，在纷繁的activity的attribute，和intent的flag中，解脱出来

让你进一步看清楚google android doc上面，究竟说的是什么

2.你如何才能杀死一个process

>= Process.killProcess

在未来，这篇文档会被更新，当有新的内容被添加

# 正文

## task是什么

你在界面上看到的任何一个activity都属于一个task，即是，你启动了一个activity，那么，它被放入到了一个task中

task维护了一个历史队列（back stack，从字面上，我们可以看到，它是一个堆栈，是为了back操作，或者是finish操作），当你点击back或者activity.finish时，task就会将旧栈顶弹出（并立刻执行销毁），转而显示新栈顶

当你长按home按键的时候，你看到了一个刚刚运行过的程序列表，而这个列表的每个entry，都是一个task（所以，严格意义，指的并不是一个app，而是task）

### task的诞生

作为一个新的小结，只是再稳固一下基础

root activity指定了task，那么谁又有可能成为root activity

1.action为main，category为launch的activity

2.使用“flag为new\_task的intent“来启动的activity

3.launchmode为singletask/singleinstance的activity

4...

task的标识，taskaffinity：

每个activity都可以设置该属性，但是，只有可以被定义为root的activity的设置才具有意义

那么其他activity呢？他们的taskaffinity被忽略，而被简单的添加到task里面

添加activity到task中的步骤：

1.是否可以是root activity?

2.对应的task是否存在？不存在，就创建一个，即是：如果存在，并不会再创建一个

3.将activity放入到task中

### back

task是一个back stack，本质上为了back操作或者finish操作，即是：弹出栈顶

设app对应的task为：task1,task2,task3

当你一直在点back，则task3的栈顶被依次弹出，当task3最终为空时，自动转到task2，然后开始弹task2的，同理，task1的

若app再无对应的task了，那么它over了（而进程出于下次启动快的目的，仍未销毁，只是优先级变低）

### task和process

#### one task & multi-processes

当你在编写特定领域的应用app1，如果你需要第三方的功能app2，请使用intent将该app2链接进来

则，运行在不同process上的activity可以被放入到同一个task中，开发者可以专注于自己的领域，并将第三方的程序“链接”进来，用户也因此感觉到了一种“无缝的一致的”体验

#### one process & multi-tasks

一个程序，可能由几个不同的task构成，设task1, task2, task3，task本身好像也存在了一个历史队列：当点击back时，程序会将task3的栈顶依次弹出，然后task2的，然后task1的，然后， 程序结束。也即是：你永远都不可能出现下面的情况：task3中还没有弹空，task2就开始弹栈顶了。

## Process

android中的每个app均运行在进程中，进程为分配资源的最小单元

通常情况下，每个app所处的process都不相同

### IPC

inter-processes communication

android提供了基于binder机制的ipc通信，也提供了aidl封装以简便开发，具体，参见：[java层aidl细节\_ipc & ibinder-binder & aidl](http://blog.csdn.net/qlsusu/article/details/6077634)

当然，你也可以认为下面的方式，也可以达到阉割版的ipc

1.intent（数据的跨进程交换）

2.content provider（数据的跨进程交换）

3..

### 杀死一个进程

因为是task引发了这一话题，所以，我们将侧重于聊聊，它和task之间的关系

#### home之后的kill

设，当前app，你看到的activity3属于task3，而整个历史task队列的形态为：task1, task2, task3

你可以按一下home按键，回到桌面，然后通过DDMS，将process杀掉，然后再重新启动app

你也许希望看到app的第一个activity，但是，很不幸，你看到的是activity3，即是：与app相关的最后一个task的栈顶

然后，你一直在按back按键，你希望看到task2, task1

但是，很不幸，当task3被清空了，app便结束了（这时，你如果再启动app，你将看到task1，因为你启动了root activity(main, launch)的task，而不是task3）

这个话题有什么意义吗？

在开发应用的过程中，你需要处理一些异常情况：

1.比如，你需要模拟home点击，就像用户一样

2.用DDMS杀死进程，就像该进程的优先级变低，最终因为其他app需要资源而被android系统回收

3.当然，紧接着，你还需要再重新点击程序，就好像用户一样，或者，像android系统在某个时刻重新启动了该app

你需要确保上述的流程，都不会出现异常，或者，程序崩溃

针对于service

如果进程启动了service，按home到桌面，在ddms上kill该进程（此种情况：kill之前没有关闭service），不久后，service重启（360的清除：比较牛逼，清理后，不会启动任何的service）

#### 显式的kill process

如果在程序中显式的调用Process.killProcess会如何呢？

设当前的界面为activityx，在调用process.killprocess之前，程序中的task为：task1,task2,task3

当调用了process.killprocess，android会判断activityi被back后，程序是否还对应着tasks

a.如果不是，那么进程被杀死

b.如果是，那么立刻启动进程，就好像要执行 类似于activityx被back的操作 一样

如果进程有显示的activity，在DDMS上kill进程，会重启