Android多进程开发Progress

Android studio查看进程：adb shell "ps|grep 你的筛选条件(比如com)"

好处：

|  |
| --- |
| a、不会影响到主业务的代码的稳定运行  b、还不占用主业务进程的内存  c、自己独立的生命周期，这样主业务进程就算被干死，也不会影响自己  d、推送进程就算崩溃了，主业务进程不受影响嘛，上面已经说过了，同第a点  e、独立进程有操作系统开辟都独立内存控件，不会和其他进程产生占用内存，影响内存分配问题，同第b点  f、独立进程的启动和退出可以完全不依赖用户对应用的使用，可以独立启动、退出，也可以不会因主业务进程退出而被结束了进程，同第c点。 |

什么情况下使用多进程： 多进程可以达到某个地方挂掉了而不影响整个app运行

|  |
| --- |
| 就我个人开发实践中就多次使用了Android多进程机制，如项目中的推送业务开发，提出要求如下：  - 不能影响主业务的代码稳定运行  - 不能占用主业务的进程内存  - 不受主业务进程生命周期影响，独立存在和运行  要满足这3个需求下，不由就会想到在应用内开辟一个新进程单独给推送业务使用，因为其特点明显:  - 独立进程运行出现了崩溃和异常而退出并不会影响其他进程运行  - 独立进程意味着有系统开辟的独立内存空间，不会和其他进程产生占用内存，影响其内存分配问题  - 独立进程的启动和退出可以完全不依赖用户对应用的使用，可以独立启动、退出，也可以不会因应用退出而结束了进程。  可以看出明显满足上述要求，那如何开辟呢。Android系统中使用多进程配置还是挺简单的，只要在Manifest中组件（如Service、Activity）直接配置android:process=“”属性即可完成配置。 Android下的多进程使用虽然很简单，但是如果不注意细节就很容易出现一些细节上的问题。 |

获取当前进程名：

|  |
| --- |
| **void** getJinc(){  **int** pid = Process.*myPid*();  String processName = **""**;  ActivityManager activityManager = (ActivityManager) getSystemService(***ACTIVITY\_SERVICE***);   List<ActivityManager.RunningAppProcessInfo> list = activityManager.getRunningAppProcesses();  **for** (ActivityManager.RunningAppProcessInfo runningAppProcessInfo : list){  **if** (runningAppProcessInfo.**pid** == pid){  processName = runningAppProcessInfo.**processName**;  }  }   Log.*i*(**TAG**,**"当前进程名称:"** + processName); } |