



链滴

# Android-内存回收

作者: [wind](#)

原文链接: <https://ld246.com/article/1449054277751>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

自己写Handler的时候，以前就一直直接这样写：

```
public static Handler mHandler=new Handler();
```

其实一直以来都知道这样写不对，最起码Google官方是不建议这样写的，它的建议写法是将mHandler作为一个内部静态类，将所在的Activity作为弱引用放到静态类中，原因是Handler的作用是处理MessageQueue中的Message，MessageQueue是个消息队列，如果消息队列中存在对UI的更新操作，那在Handler对象处理这一Message之前，UI所在的Activity都不能被释放，这就有可能造成内存泄露。下面是正确的写法：

```
public static class MyHandler extends Handler{

    `WeakReference<MainActivity> mActivity;`
    `MyHandler(MainActivity activity){`
        `mActivity=new WeakReference<MainActivity>(activity);`
    `}`
}
```

之前就有接触过弱引用软引用这些东西，现在正好趁这个机会来了解一下android的内存回收过程：

首先，Android基于进程中的组件及其状态规定了五个回收优先级，按顺序为Empty Process, Background Process, Service Process, Visible Process, Foreground Process。当内存不足需要释放时，Android会按顺序查找其可以释放的Process以便优化系统，那么具体的过程是怎么样呢？

1、Android的回收触发点大概有以下几个：

(1) 用户启动新的Activity或者应用程序，开启一个新的进程 (2) 用户按back，退出当前应用程序

当触发点被触碰，系统会调用相关的函数如下。

2、获取当前系统中不可删除的进程，包括运行Service的进程，和系统自带的进程（属性android:persistant="true"）

3、获取当前系统允许的最大进程数以及当前进程数，如果有需要释放空间则需满足以下条件：

(1)非系统进程

(2)不带Service，BroadcastReceiver的进程，空进程

4、按3所说条件释放后，若还不满足条件，则放宽条件以释放内存，包括一些带Service的进程。

那么我们知道，其实很多app在手机是常驻的，无法停止它。这是为什么？之前我上网搜，网上答案多，比如添加对系统各种反应的Receiver，给app程序加上persistent属性，但其实打开手机，查看在运行的app时，我们都可以发现常驻的app都有个特点，就是一般有两个以上process，包括之前曝的蜻蜓FM应用，之前发现了其实不太明白或者说不太懂他们之间的机制是怎么样的，后来看到一本讲android源码的书，看了Binder的IPC通信，后来发现process在jni层是可以自行创建的，也就是说，app背后存在两个process，相互通信，有一方被杀死，死之前通过IPC通信告诉另一个app进程，通过一个进程再重新启动这一进程，这样就实现了app永远不会挂掉的情况。

关于android的强弱引用之前看书的时候也写了总结，但是自己不做系统级开发的话，可能一直都不透彻吧，希望有机会可以做做Android ndk的开发。