**Android 应用程序启动过程**

参考：<https://blog.csdn.net/Luoshengyang/article/details/6747696>

1. Android应用程序进程的入口函数是ActivityThread.main。应用程序进程创建完成后，Android应用程序框架层就会在这个进程中将ActivityThread类加载进来，然后执行它的main函数；
2. 在系统启动的过程中SystemServer为ActivityManagerService创建一个独立的进程，Android应用程序的进程由ActivityManagerService负责创建；
3. ActivityManagerService.startProcessLocked()调用Process.start()为应用程序创建新的进程;在Process.start()中传入"android.app.ActivityThread"加载ActivityThread并执行其main()函数；
4. Process.start()调用Process.startViaZygote()将创建进程的参数放到argsForZygote列表，"--runtime-init"表示要为新创建的进程初始化运行时库。
5. 调用Process.zygoteSendAndGetPid将argsForZygote列表通过socket传给ZygoteInit.runSelectLoopMode，通过Zygote.forkAndSpecialize创建一个进程；
6. 进入到RuntimeInit.zygoteInit()，先通过RuntimeInit.zygoteInitNative()完成Binder驱动程序的初始化工作，然后RuntimeInit.invokeStaticMain()执行进程的入口函数main()；
7. RuntimeInit.zygoteInitNative()会进入到AppRuntime.onZygoteInit()，调用ProcessState::startThreadPool启动线程池，该线程池中的线程用来和Binder驱动程序进行交互；然后调用IPCThreadState.joinThreadPool()告诉Binder驱动程序线程要进入循环了；接着调用talkWithDriver通过ioctl文件操作函数不断与Binder驱动程序进行交互，以便获得Client端的进程间调用；
8. 有了该线程池之后，当Android应用程序要和其它进程中进行通信时，只要定义自己的Binder对象，并把该Binder对象的远程接口通过其它途径传给其它进程，其它进程就可以通过这个Binder对象的远程接口来调用我们的应用程序进程的函数了。
9. 在RuntimeInit.invokeStaticMain()中，通过ClassLoader.loadClass()将ActivityThread加载到进程，通过反射获取到ActivityThread.main()，然后再抛出一个ZygoteInit.MethodAndArgsCalle异常，ZygoteInit.main函数在捕获这个异常时，通过反射调用ActivityThread类的main函数
10. ActivityThread.main()中创建一个ActivityThread对象，并进入消息循环。