享学课堂诚邀作者：周周

转载请声明出处！

前言

手把手讲解系列文章，是我写给各位看官，也是写给我自己的。
文章可能过分详细，但是这是为了帮助到尽量多的人，毕竟工作5,6年，不能老吸血，也到了回馈开源的时候. 这个系列的文章：
1、用通俗易懂的讲解方式，讲解一门技术的实用价值 2、详细书写源码的追踪，源码截图，绘制类的结构图，尽量详细地解释原理的探索过程
3、提供Github 的 可运行的Demo工程,但是我所提供代码，更多是提供思路，抛砖引玉，请酌情cv 4、集合整理原理探索过程中的一些坑，或者demo的运行过程中的注意事项 5、用gif图，最直观地展示demo运行效果

如果觉得细节太细，直接跳过看结论即可。
本人能力有限，如若发现描述不当之处，欢迎留言批评指正。

学到老活到老，路漫漫其修远兮。与众君共勉！

引子

app性能优化，是每一个高阶开发者必备技能，但是现在网络上关于性能优化的文章大多即没有成文案例，又没有知识体系介绍，让需要了解此项技术的人无从下手学习。本系列文章，将先讲解案例
，让你看到效果，再详解细节，让你知晓原理。希望对大家有帮助。

案例

启动时间优化

app性能优化，自然是存在问题，然后才优化，那么如何去诊断出这些问题呢？自然是有手段。

Q：我们如何得知我们自己的app启动花费了多少时间？ A：adb 命令。步骤如下：

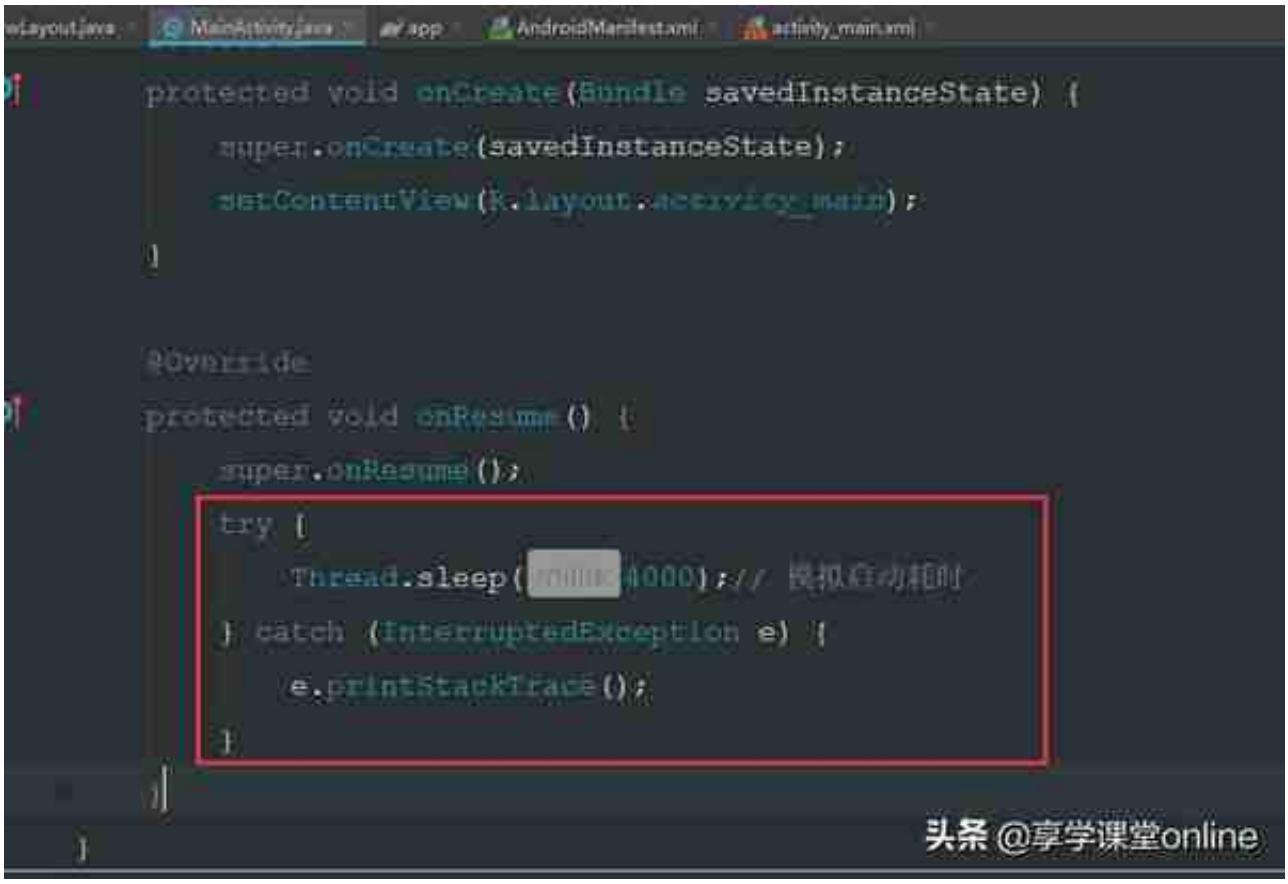
- 确保设备连接到电脑；
- 启动cmd窗口
- 输入如下命令：adb shell am start -W [app包名]/[launcherActivity的全类名] 在android29模拟器上的结果为：

这里会出现3个time： thisTime：am start 命令可能会启动多个Activity，如果启动多个，thisTime则是指最后一个Activity的启动时间，如果启动的是1个，那么thisTime等于TotalTime. TotalTime：新的应用的Activity启动的耗时。 WaitTime: AMS将当前Activity从onResume转向onPause，再启动新应用Activity的总时长，包含了TotalTime在内，所以WaitTime比TotalTime要长。

当然你也可以加上-S -R 10 ，连续启动10次，然后自己计算平均启动时长。

adb shell am start -S -R 10 -W packagename/.MainActivity

这里的3个时间，我们大概可以看出自己的app启动具体花费了多少时间。通常启动时间可以通过肉眼观察得到，但是具体到确切数值，还是需要借助命令行的。那么接下来的问题，如果发现app启动耗时不理想，比如非常极端的情况，我们在Activity的onCreate中加入了一些耗时操作，


protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
}
@Override
protected void onResume() {
super.onResume();
try {
Thread.sleep(4000);// 模拟启动延时
} catch (InterruptedException e) {
e.printStackTrace();
}
}
头条 @享学课堂online

那么：