

# 低内存导致webview被杀

---

webview最终崩溃的原因是由于运行内存达到了一个阈值

## 在android中如何查看webview版本

1. 打开手机的设置（Settings）应用。
2. 在设置应用中，向下滚动找到并点击或点击进入 "应用"（Apps）或 "应用管理"（Application Manager）选项。
3. 在应用列表中，查找并点击 "Android 系统 WebView" 或 "Webview"。这可能在 "全部应用"（All Apps）或 "已安装应用"（Installed Apps）选项卡中。
4. 进入 WebView 应用信息页面后，你可以在这里找到 WebView 的版本号。它通常在应用名称下方或其他关于 WebView 的详细信息部分。

通过上面的操作，我们得出一个结论

webview的版本提升通常是跟随操作系统的，在不同app中，开发者使用的webview是由系统提供的，开发者可以通过hack的方式重新安装测试机的webview软件程序

## webview占用系统内存的组成

webview总内存= native内存+gpu内存

native内存包括：v8、BlinkGC、PartitionAlloc、DiscardableSharedMem、SharedMemory、malloc、fontcache、web\_cache。

- v8：V8 Javascript 引擎所管理，一般用于分配 Javascript 对象和数据。
- BlinkGC：是标记式垃圾回收算法堆，一般用于管理页面运行上下文对象。
- PartitionAlloc：为分桶式内存分配算法，用于解析、排版、页面运行上下文以及临时内存。
- malloc：为类 libc 堆实现的仅用于 浏览器内核渲染引擎分配算法，缓存、页面运行上下文、临时内存都有用到。
- SharedMemory：为多进程共享内存。
- DiscardableSharedMem：一般为图片解码缓存和 GL资源缓存等所用。

gpu内存包括：

- gpu、skia。

## 优化方案参考

1. 降低gpu内存占用 结合业务做出抉择，类似于分布式的base理论的抉择
  - 一定在屏幕内的多个图片，不会通过滑动而影响显示的，可以试一下合成一张图片，减少图片内空白区域的占用，达到降低内存的目的
  - 超出屏幕之外的任何元素，不要保留任何引用，进行删除操作，或者使用虚拟滚动
  - 避免内存泄露
  - 尽可能的减少标签创建的数量来达成同样绘制的目的

- 图片压缩，多图片地方尽量使用缩略图
- 避免使用非常大的canvas以达到"高清的目的"
- 禁止使用canvas的webgl模式
- 设计角度上减少背景图的使用，尽量以css绘制简单元素为主
- 流程上尽量采取进入新页，释放旧页的方式
- 任意图片的懒加载也是有效降低瞬时内存的方式
- 对于css动画，应该减少动画的时长，帧率，复杂性，同时使用动画加速。

## 2. 降低js代码的内存占用

- 降低虚拟dom的缓存量，切换路由尽量不对虚拟dom进行缓存，尽量从业务设计上解决这个问题
- js堆内存的申请导致内存泄露问题，减少大的全局变量的使用，尽量在函数中，同时也要注意闭包带来的内存泄露问题
- 禁止使用web worker多进程操作

## 3. webview层优化

- webView reload
- 避免使用多进程

## 参考

- [淘宝webview优化（提高淘宝应用的性能和用户体验）](#)
- [深入探讨如何从测试的角度对Webview进行压测并精准地分析Webview的内存情况](#)
- [WebGL与Canvas的显存与内存使用分析](#)
- [史上最全的WKWebView问题优化指南](#)
- [CSS频繁绘图中transform导致的内存占用问题](#)
- [WebView 常见 Crash 分析及解决方案](#)