

# iOS之 Crash log

对于已发布的 APP，根据出现问题时的闪退日志，分析和判断 Crash 的原因，快速准确的定位和解决。

## crash log

---

### 概念

在 iOS 上运行的 APP 出现 Crash 的时候，通常会生成一个 Crash log，记载问题发生时的具体状况。

- 开发者可以在 iTunes Connect 中特定 APP 下找到收集上来的 Crash log。
- 也可以连接电脑，去本地目录找

Mac :~/Library/Logs/CrashReporter/MobileDevice/ 这个时候你会发现一大堆的.crash 文件和.ipsw 文件。

- (推荐)通过 Xcode 获取到崩溃日志，方法是 Xcode->Window->Devices，想必很多开发者和笔者一样，也都是用的是这个方法。

### Crash log 分析

通过上面三种方法收集到了 Crash log，但用文本编辑器打开文件是一堆十六进制的内存地址。

Crash log 里面包含了 Crash 发生的 APP、运行软硬件环境、发生时间、错误类型、方法调用异常栈、各线程状态、寄存器和内存信息。

意义最为重大的，可能就是异常线程的调用栈。可惜有些时候，这关键的信息竟然全是 16 进制的数据，所以我们很难看懂。如：

```
1CrashDebugInfoTest 0x1000c2b90 0x1000bc000 + 27536
```

要从十六进制的地址码，得到我们代码中对应的方法调用，就需要结合调试信息对 Crash log 进行符号化。笔者查看了很多文档，能符号化 Crash log 无非就 3 种方法：

1. 使用开发工具库中自带的 symbolicatecrash
2. 使用 atos
3. (推荐)使用 dwarfdump

国外的比如 New Relic，还有国内 OneAPM 的[Mobile Insight](#)。