

Swift 2: 拥抱变化

Kimmy Leo <***@o**k.com>

前段时间苹果公司 WWDC 2015 上，Swift 作为一个简单的点一闪而过。相比起来，大部分人更关注的是 HomeKit、El Captain 和 watchOS 等更新；但作为一个新型的编程语言，Swift 2 的更新和宣布开源，在程序员圈子里面还是引起了不小的轰动。

近年来，编程语言成了各大 IT 公司布局的重点，近日 Mozilla 刚刚发布了 Rust 的 1.0 正式版本，微软的 TypeScript 也在不断的更新升级中，同样的还有 Facebook 在去年发布的 Hack 以及 JetBrains 的 Kotlin 等等。苹果在这个时间宣布 Swift 开源，也有迎合大势的意思在。

作为一个编程语言新星，Swift 集众家之长，大胆地引入了 Abstract Data Type、模式匹配以及类型推导等特性；同时采用了使用范围比较广的一些语法结构（比如 class、struct 关键字以及花括号代码块等）；另外值得一提的是，Swift 生成的是面向 llvm 的代码，无疑给跨平台带来了很多的好处，也使得开源之后的应用会更加广泛。

当然这并不是 Swift 能够受到热切关注的主要原因。Swift 广受关注，最主要的还是因为苹果公司这个后台做支撑，以及苹果旗下的各种平台和产品做应用。同时，相较于苹果平台上之前较流行的 Objective-C 来说，Swift 在各方面都比较有优势：语法简洁清晰，同时又多出了泛型、模式匹配等强大的功能，而且可以轻松的跟原有的 Objective-C 代码交互，也就是说，在几乎不用改动代码的情况下可以轻松的将 Objective-C 的原有项目迁移到 Swift。

由此我们可以看出苹果在 Swift 上投入的精力和期望，很显然，苹果是希望通过 Swift 来逐步替换掉 Objective-C，成为其平台开发的主要语言。或者更进一步地说，苹果希望把 Swift 做成一个自己的产品。一方面，产品化可以是他们能够控制主要技术及发展方向；另一方面，对开发者来说，产品化会形成一个稳定的技术生态圈，使开发者更多的关注业务和产品创新。

但是，回头来看 Swift，其实还是有很多问题。最重要的就是由于其最初是设计用于苹果平台上的应用开发，所以对苹果平台的依赖（Foundation 和 Cocoa 库）十分严重。同时因为其继承自 Objective-C 的特性（如 ARC 和 GCD）会带来一些运行时依赖，也给 Swift 进行系统级编程带来一定的难度。所以，Swift 的定位应该与 Kotlin、Scala 等相似，被作为一个拥有高阶抽象的应用开发语言来使用。

我曾在知乎的一个答案里面调侃 Swift 目前的能力很弱。其原因很明显，作为一个新兴语言，想要从其现有的领域突破，成为一门通用语言的话，所需要做的不仅仅是添加特性和开源这么简单。虽然有了苹果作为后盾来支撑其发展，Swift 成长的道路依然漫长。

要成为一个通用语言，Swift 仍需要一步步地完成下面几项任务：

1. 实现一套跨平台的标准库。这也是首要的任务。Swift 目前对苹果自家的库依赖严重，而缺少可跨平台的标准库。虽然 Swift 已经在语言层面对文本处理、集合框架进

行了支持，但作为一门通用语言，这是远不够的，文件系统、IO 和网络操作等常用的基础设施都需要提供支持。

2. 实现通用的外部功能接口 (FFI)。对于一个新生语言来说，这一点也很重要。很多比较通用的库已经有了跨平台实现，如果有完善的 FFI，只要把这些已有的库用 Swift 进行简单的封装就可以使用了。这样大大减少了重造轮子的机率，并且能够很好的扩展 Swift 的适用范围和使用场景。
3. 实现自动构建及依赖管理工具。对于编程语言来说，这是目前的一个趋势。Swift 跨平台以后必然要面对其他平台上没有 Xcode 这种量身定做的 IDE 的局面，而自动构建和依赖管理工具可以简化 Swift 的项目创建、构建、测试和发布等各种工作，提供一套自动化的工具链来补充替代 Xcode。
4. 实现语法、语义检查分析工具。同样地，这也是弥补脱离 Xcode 以后，在其他平台下进行高效地 Xcode 编程的一套工具。通过语法和语义分析，便于在没有 Xcode 的情况下，使用其他的编辑器/IDE 进行静态语法检查和智能提示等操作，来提升开发人员效率。
5. 构建一套或多套通用应用框架，完善技术生态圈。这一点是 Swift 所要完成的核心任务，也是目前广大开发者们所关注的重点。也只有通过这个，才能够回答“Swift 开源以后能做什么”这个问题。当然，如果前面的工作做得足够好的话，这项任务也可以由开源社区来完成。

Swift 在发布之初就寄予了厚望，而且纵观其语言特性和发展趋势，可以看出整体表现还是非常令人满意的。如果在开源以后能够做得更好，那么也许在诸如分布式计算、服务器构建以及 Web 开发等领域成为主流。这样回头看，不得不感叹苹果的用心之深。