

25 | 桌面开发的未来

2019-07-16 许式伟

许式伟的架构课

[进入课程 >](#)



讲述：姚迪迈

时长 11:37 大小 10.66M



你好，我是七牛云许式伟。

今天，我原本想结合一个实战例子，来回顾一下“桌面软件开发”一章到今天为止的内容，但是由于要准备的内容比较多，会延后一下。

所以，今天我还是会按原来大纲的内容，谈谈我个人对桌面开发未来趋势的判断。

桌面平台的演进与未来

谈未来，我们要先看过去。

在 PC 时期，本地桌面操作系统主流的有 Windows、MacOS、Linux。为了消除不同平台的差异，于是就出现了 QT、wxWidgets 这样的跨平台解决方案。

但是它们都败了，败给了一个它们并没有意想得到的对手：**PC 浏览器**。

浏览器并不是为跨平台而来，但是除了干成了软件服务化外，也干成了跨平台这件事情。

虽然浏览器厂商很多，但是它们遵循相同的规范。**这意味着支持了浏览器，就支持了所有的 PC 用户。**

这太诱人了。

于是在软件服务化和跨平台开发的双重优势下，软件厂商们趋之若鹜，QT、wxWidgets 这些方案就变成小众需求。

QT 有机会反抗么？其实是有的。关于这一点我们后面再说。

然后，移动大潮来了。我记得 2006 年有一次我和雷军雷总吃饭，聊起对移动操作系统未来趋势的判断，我们俩各持己见。

雷总认为 WinCE 会赢，因为 Windows 已经培育了最大的开发者群体。而我认为 Symbian 会赢，因为它占据了最大的终端用户群。

结局大家已经知道了，最后赢的是谁都没有预料到的玩家：Android 和 iOS。

如果我们从事后看，实际上这个事情并不是完全没有迹象可循。iOS（诞生于 2007 年）固然当时还没有诞生，但是 Android 诞生于 2003 年，并于 2005 年被 Google 收购。作为搜索引擎厂商，Google 收购一个手机操作系统，显然不是随意为之的，而是公司发展战略上的考量。

Android 和 iOS 的诞生，一下子让操作系统的生态变得更为复杂。

操作系统不同，输入方式不同（鼠标 vs 触摸屏），屏幕大小不同，想要一套代码横跨 PC 和移动多个平台？太难。

这还不算。虽然还不像手机那么普遍，但是今天手表、电视机、汽车，以及各式各样的 IoT 传感设备，都需要操作系统的支持。

从操作系统发展来说，我个人会倾向于按交互方式来分。未来桌面操作系统和服务端操作系统会渐行渐远，差异越来越大。**从交互来说，服务端会维持简约，经典的命令行交互会长期占据主流。**

而桌面操作系统，笔记本市场，鼠标 + 键盘仍然会占据主流。虽然鼠标形态已经变了，变成了触控板，但是鼠标指针这种基于精确位置交互的方式会得到保留。多点触摸的交互，也会得到部分机型的支持。

移动市场，多点触摸 + 键盘会占据主流。但是语音助手也会得到一定程度的渗透。

IoT 市场，语音助手会占据交互的主流。但也会有一些设备引入多点触摸这种交互方式来补充。在这个市场，目前看技术上的 Amazon 和 Google 占据了领先地位。虽然苹果入场较早，但是 Siri 的表现还是和前两者有较大的差距。

IoT 设备会两极分化。**一类 IoT 设备是专用设备，它的应用场景非常固定，它对操作系统最大诉求是裁剪能力：最好不要给我太多的东西，匹配场景就好。**能不能提供 AppStore？不是重点，有也只有很少的一些应用，其实直接找合作伙伴就好。

一类 IoT 设备则有较大的通用性。但受限于语音助手技术的限制，IoT 操作系统的开放性要比移动系统差很多。所以在有任何可能的时候，这些设备就会带上触摸屏变成一台由移动系统支持的设备。

长远来说，要看智能语音技术的发展。关于这一点，我个人抱谨慎乐观的态度。但显然，在很长一段时间里，我们面对的还是移动操作系统。

这么多操作系统怎么搞呢？

于是 React-Native 出现了。理论上，React-Native 可以横跨 PC 和移动。因为 React 本身基于 Web 技术，可以支持 PC 浏览器，而 React-Native 又支持 iOS 和 Android，从而做到 “Learn once, write anywhere”。

平台差异不能完全消除，只能尽可能地减少。

手机操作系统这场仗刚有了眉目，移动浏览器之争又起来了。

国内涌现了大量的小程序厂商，国外 Google 也在推 PWA。还有 Facebook 意见不明，不知道会去支持 PWA，还是基于自己的 React-Native 技术搞一套新的移动浏览器标准。

这下好了，统一的 Web 分裂成多个技术阵营。

移动浏览器，国内外不统一已经是既成事实。海外巨头们除了 Facebook，已经用明确的行动支持 PWA。小程序在海外要想有市场，要看头条腾讯阿里们的海外市场占有率。

移动 WebApp 技术的分裂是否会最终得到纠正？这仍然是未知之数。

但由此观之，终端操作系统的多元化已经是既成现实。这对开发者生态将产生重要的影响。

我们可能有人留意到，QT 今天基本上支持了所有的桌面操作系统，不管是 PC 还是移动。但是这还不太够，因为还差 Web、小程序和 PWA。

今天的跨平台，重点是要跨 Android、iOS、Web、小程序和 PWA。如果精力顾不上，PC 桌面操作系统的优先级反而可以缓一缓，毕竟 Web 也能够顶一下。

QT 的机会在这里。但是很明显它并没有意识到兼容 Web 开发对于一个跨平台工具的重要性。

就算在 PC 时期，一个同时支持 Web 和本地操作系统的跨平台工具也能够受到欢迎。今天随着桌面平台的多元化，跨平台工具的需求达到了历史最高点。

当然还有一种跨平台的思路，是垂直发展，比如专做游戏开发的跨平台。不过单就游戏开发这个领域而言，已经有强大的玩家，比如 Unity 在里面。

那么，通用的跨平台怎么做到？

Google Flutter 给了一条路，它把对操作系统的要求最小化，整个界面系统完全自己在用户态构建。

这个思路和 Go 语言有点像。Go 语言其实是在用户态完全重写了操作系统的进程管理和 IO 子系统。

那么 Flutter 会像 Go 语言一样成功么？

我个人持谨慎态度。不同操作系统的用户是有自己独特的交互范式的。比如 Android 和 iOS 用户的习惯就有一定的差异。而这可能恰恰是跨平台更难的一点。

另一个是软件体积问题。Android 是 Google 自己的，可以通过让 Android 预装基础库来减少体积。但是更多的系统有可能需要一个体积不小的跨平台层。

这会制约 Flutter 的发展。客户端软件的尺寸，对新用户的转化率有着至关重要的影响。何况像微信小程序这样的平台，还限制了小程序的尺寸，最早限制为 4M，后来放宽到 8M。

这和 Go 语言面临的环境不太一样。Go 语言因为面向的是服务端，用户对软件的尺寸不敏感，反倒是部署的便捷性更敏感。

我个人更倾向于尺寸更轻盈的跨平台工具。

其次是编程手法上的问题。大趋势是要用 Web 这种声明式的界面描述方式。至于是否需要在语法上进行一次重新梳理，我个人觉得是有必要的。React-Native 在这个方向的探索是个不错的尝试。

在这一点上，苹果的 SwiftUI 或许更值得关注。苹果以极简体验著称，SwiftUI 某种程度上来说代表了关于跨平台开发的可能方向。

儿童编程教育

在我们谈论桌面开发的时候，我认为其实还有一个重要但又很容易被忽视的趋势，是儿童编程教育的走向。

说到儿童编程教育，我们大多数人可能都知道 Scratch 语言。但是要说儿童编程的鼻祖，毫无疑问应该算 Logo 语言，海龟作图。

Scratch 语言由美国麻省理工大学（MIT）于 2007 年发布，到现在已经发展到了 3.0 版本，项目正变得越来越活跃。

在 Scratch 之后，Google 也曾经发布了 Blockly 语言进军儿童编程教育。但是由于缺乏社区基础，Blockly 语言一直不温不火。

但有两件有趣的事情。

其一，Scratch 3.0 是基于 Blockly 的源代码改造而成的，为此据说 Google 也投入了大量的技术人员进行协助，双方协同开发。

其二，Google 基于 Blockly 语言搞出了一个 App Inventor，用于教育儿童学习 Android 开发。

无独有偶的是，苹果推出的 Swift 语言启蒙教程也是针对儿童的，在 AppStore 上可以下载到，叫 “Swift Playgrounds”。

这意味着，我们原本以为两件风马牛不相及的事情，其实是密切相关的。

桌面开发的未来是什么？

从终局的视角来看，桌面开发的终极目标，是让儿童可以轻松编写出生产级的应用。

这不是痴人说梦。

在 iOS 出来之前，如果有人说他要开发一个让三岁小孩都会使用的电脑，可能会有很多人觉得绝无可能。

但是苹果的确做到了。虽然可能还不能完全识别电脑上常见的文字，但是一个三岁的儿童使用起 iPhone 或者 iPad 却毫不困难。

那么，让一个八岁刚刚上学没多久的小学生去做生产级的应用，这事也不是遥不可及的梦想。

桌面开发技术的演进，和儿童编程教育相向而行，有一天必然汇聚于一点上。

结语

今天我们结合桌面开发和儿童编程教育，聊了个人对桌面的未来演进趋势的判断。

如果你对今天的内容有什么思考与解读，欢迎给我留言，我们一起讨论。下一讲我们将结合一个实际的案例，来讲解一下桌面开发（含单机软件和 Web）到底是什么样的。

如果你觉得有所收获，也欢迎把文章分享给你的朋友。感谢你的收听，我们下期再见。



许式伟的架构课

从源头出发, 带你重新理解架构设计

许式伟
七牛云 CEO



新版升级：点击「 请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 24 | 跨平台与 Web 开发的建议

下一篇 26 | 实战(上)：怎么设计一个“画图”程序？

精选留言 (15)

写留言



大王叫我来巡山

2019-07-16

每一篇都读很多遍，深度是需要自己的努力的，但是方向还是需要老师来带，好的专栏不局限于传授知识，更多的是交流思想，感谢许老师

展开 ∨



6



小泥人

2019-07-16

桌面跨平台Electron呢？

展开 ∨

💬 1

👍 3



有铭

2019-07-16

编程技术的进化方向就是尽量下沉，让更多“水平一般”的开发者甚至是儿童，能写出在80%场景下可以在生产环境下跑的程序，剩下20%的问题交给专家解决。这个思路在上世纪60年代就被确定下来了，毕竟水平越低的开发者薪水越低，节约成本。所以我的感觉就是，未来IT这块，除非你真打算做研究，爬格子，写论文。否则的话，广度比深度优先，广度能让你迅速的用已知轮子快速的组装出系统来验证业务。而研究深度，如果运气不...

展开 ∨

💬

👍 2



盛

2019-07-16

受教了：视角完全不一样，每次看许老师的东西都觉得能够换个角度和方式去思考一些技术相关的事情。

展开 ∨

💬

👍 2



choes

2019-07-17

最近我也在教我8岁的外甥用Scratch编程，他称这个叫猫猫软件，感觉就是搭积木来实现自己的想法，完全想象不到有一天儿童也能做生产级的应。之前的文章里许老师有说过教小孩编程，请问也是用Scratch吗？还是用Python或Go之类的语言呢？

展开 ∨

作者回复：从scratch开始的，但是不限于它



💬

👍 1

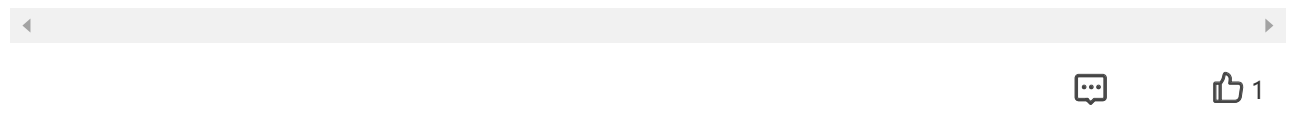


默片。

2019-07-17

qt的qml和webassembly支持是否可以从某种程度上看做对web的支持呢

作者回复: 嗯，webassembly算



二马

2019-07-19

许大大如果写科技史，科技人文类的专栏或者书，相信也会和吴军博士的《浪潮之巅》一样精彩！



kirogiyi

2019-07-17

曾经想过软件开发的未来，大部分软件开发工作可以通过机器学习进行机器编程，人类只需要做好模型设计和下达指令就可以得到自己想要的软件，逐步取代如今大部分靠人工开发软件的方式，从而降低人工成本和提高效率(当然也意味着未来软件开发的门槛更高——精英编程)。现在许老师讲到了儿童编程的未来，那么儿童编程在未来是否会催生一种新的软件开发方式或者改变IT行业的整个布局？

展开 ∨



Charles

2019-07-16

许老师能概括下，你是怎么教儿童编程的吗？目标已经是让他做生产级应用了吗？谢谢



王克

2019-07-16

以前记得微软还是Google想做一件事情，通过超级浏览器集成各种软件，让PC使用者不再安装软件，只需要一个浏览器就完成基本的需要，后来证明这路没有走通。我想大概是因为需求千变万化，一个集中的工具完不成所有的事，而且没有形成标准，开发者加入不了。现在PC的H5推出，兼容web，小程序，移动端，很强大。对于H5，不知道你怎么看？

展开 ∨

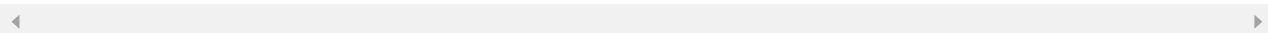


Geek_88604f

2019-07-16

如果小孩也能写出生产级的软件，那么码农的未来在哪里？

作者回复: 挺好的问题，可以问问自己这个问题



1



lckfa李钊

2019-07-16

和许老师的观点不谋而合，桌面开发的未来就是少儿编程的未来，未来一定会有更小的开发者入场，所以抢占少儿编程的先机其实就是在深耕未来。

展开 ∨

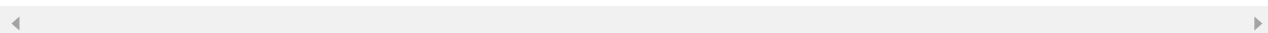


antipas

2019-07-16

能谈谈您对Fuchsia的看法么，它的优劣？它会是IoT时代的操作系统么

作者回复: 还没有研究过



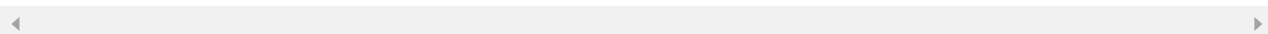
Geek_88604f

2019-07-16

浏览器是如何做到跨平台的，许老师？

展开 ∨

作者回复: 和 Flutter 类似





Aaron Cheung
2019-07-16

趋势 未来 打卡25

展开 ∨

