

## 06 | 操作系统进场

2019-05-03 许式伟

许式伟的架构课

[进入课程 >](#)



讲述：姚迪迈

时长 11:46 大小 10.78M



你好，我是七牛云许式伟。

在编程语言出现后，软件生产效率得到了大幅度的提升。随着越来越多软件的出现，自然而然就诞生了多个软件如何共处，也就是软件治理的需求。比如下面的这些需求场景。

多个软件如何同时运行（多任务的需求）？

多个软件如何共同使用计算机上的存储空间（内存管理、文件系统的需求）？

多个软件如何共同使用同一个外部设备（设备管理的需求）？

多个软件如何相互通讯，如何进行数据交换（进程间通讯、共享内存的需求）？

病毒、恶意软件如何治理（安全管理的需求）？

如果没有一个中间的协调方，软件与软件之间并不知道彼此的存在，你不难想象出，这种没有统一规则约束下的场面，会有多么凌乱。

于是，操作系统就出现了。对于软件而言，它像一个大法官，制定规则并据此约束大家的行为。

## 操作系统的启动过程

操作系统是怎么获得执行权的？

这是计算机主板 ROM 上的启动程序（BIOS）交给它的。

计算机加电启动后，中央处理器（CPU）会从一个固定的存储地址加载指令序列执行。通常，这个固定的存储地址指向计算机主板的 ROM 上的一段启动程序（BIOS）。这段启动程序通常包含以下这些内容。

存储设备的驱动程序，用以识别常规的外置存储设备，比如硬盘、光驱、U 盘。

基础外部设备的驱动程序，比如键盘、鼠标、显示器。

设备和启动配置的基础管理能力。

支持执行外置存储中引导区的机器代码程序。

跳转到外置存储引导区的固定地址，把执行权交给该地址上的引导程序。

引导区的引导程序有长度限制（关于这一点我在上一节已经介绍过），只能做非常少的事情。在常规情况下，它只是简单地跳转到真正的操作系统的启动程序，但有时计算机上安装了多个操作系统，此时引导程序会提供菜单让你选择要运行的操作系统。

这样，操作系统就开始干活了。

## 操作系统的需求演进

那么，操作系统是做什么的？前面我们说的“软件治理”是否可以涵盖它完整的目标？

让我们从操作系统的发展历程说起。

最早期的计算机是大型机。这个时期的计算机笨重、昂贵，并且操作困难，主要使用人群是搞科研性质的科学家或其他高端人群。

虽然这个时期催生了 IBM 这样的硬件巨头，但大多数人根本就意识不到，这玩意儿对后世人们的生活能够产生如此翻天覆地的变化。

这个时期的计算机还是单任务的，以计算为主，软件为操作硬件服务。如果我们认为“软件治理”是操作系统的根源需求的话，那么可以认为这个时期还不存在操作系统。但的确会有一些辅助工具库来简化用户使用计算机的负担，我们可以把它看做操作系统的萌芽。

从这个意义来说，提供计算机的“基础编程接口”，降低软件开发的负担，是操作系统更为原始的需求。

此后，小型机和个人计算机（PC）的崛起，分别诞生了 UNIX 和 DOS 这两个影响深远的操作系统。UNIX 就不用说了，它几乎算得上今天所有现代操作系统的鼻祖。

DOS 的历史非常有趣。首先是 IBM 没把操作系统当回事儿，把这个活儿包给了微软。然后是微软只花了 5 万美元向西雅图公司购买了 86-DOS 操作系统的版权，更名为 MS-DOS。

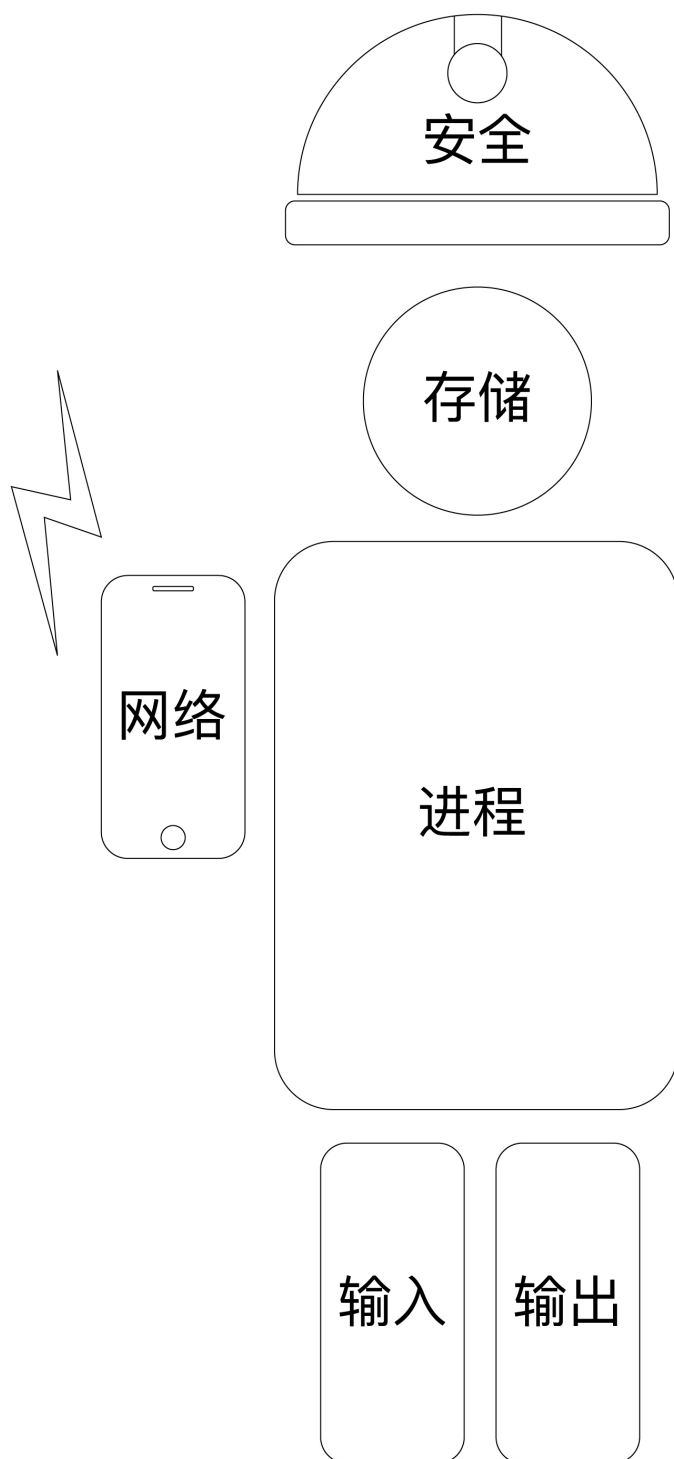
那么 86-DOS 是怎么来的？西雅图公司的一个 24 岁小伙叫蒂姆·帕特森（Tim Paterson），单枪匹马花了 4 个月时间写出来的。

可以看到，这个时期人们对操作系统并没有太深刻的认知，多数人只把它看做硬件的附属品。IBM 不把它当回事，西雅图公司也没把它当回事，几万就把它卖了。只有微软认认真真地把它当做生意做了起来（在此之前微软的生意是卖 BASIC 语言的解析器起家，所以微软一直对 BASIC 语言情有独钟，直到很久以后微软搞出了 C# 语言后，情况才有所改变）。

等到 IBM 意识到操作系统是个金蛋，改由自己做 PC-DOS 操作系统的时候，微软已经通过推动 PC 兼容机的发展，让操作系统不再依赖特定的硬件设备，微软也就因此脱离 IBM 的臂膀，自己一飞冲天了。

回到问题。要回答操作系统在做什么，我们可以从客户价值和商业价值两个维度来看。

客户价值来说，**操作系统首先要解决的是软件治理的问题**，大体可分为以下六个子系统：进程管理、存储管理、输入设备管理、输出设备管理、网络管理、安全管理等。



**操作系统其次解决的是基础编程接口问题。**这些编程接口一方面简化了软件开发，另一方面提供了多软件共同运行的环境，实现了软件治理。

商业价值来说，操作系统是**基础的刚需软件**。计算机离开了操作系统就是一堆废铜烂铁。随着个人计算机采购需求的急速增加，光靠软件 License 的费用就让操作系统厂商赚翻了。

虽然第一个广为人知的操作系统是 UNIX，但从商业上来说最成功的操作系统则是 DOS/Windows，成就了微软的霸主地位。

为什么是 DOS/Windows 赢得了市场，这无关技术优劣，关键在于两者的商业路线差异：UNIX 走的是企业市场，而 DOS/Windows 选择了更为巨大的市场：个人计算机（PC）市场。

操作系统也是**核心的流量入口**。占领了操作系统，就占有了用户，想推什么内容给用户都很容易。微软对这一点显然心知肚明。

这也是为什么当年网景推 Netscape 浏览器的时候，微软很紧张。因为浏览器是另一个软件治理的入口，本质上是操作系统之上的操作系统。如果软件都运行在浏览器上，那么本地操作系统就沦为和硬件一般无二的管道了。

虽然早期操作系统没有应用市场（AppStore），但是通过操作系统预装软件的方式向软件厂商收租，这是一直以来都有的盈利方式。国内盗版的番茄花园 Windows 发行版就是通过 Windows 系统上预装软件来盈利。

当然预装软件只是一种可能性，流量变现的方式还有很多。苹果的 iOS 操作系统开启了新的玩法，它构建了新的商业闭环：账号（Account）、支付（Pay）、应用市场（AppStore）。

我们把这个商业模式叫收税模式。帐号（注意是互联网帐号，不是过去用于权限管理的本地帐号）是前提。没有帐号，就没有支付系统，也没有办法判断用户是否购买过某个软件。

应用市场实现了应用的分发，既解决了系统能力的无限扩展问题（客户价值），也解决了预装软件的软件个数总归有限的问题（商业价值）。支付则是收税模式的承载体，无论是下载应用收费，还是应用内购买内容收费，都可以通过这个关卡去收税。

无论是本地操作系统 iOS 和 Android，还是 Web 操作系统（浏览器）如微信小程序，都实现了“帐号 - 支付 - 应用市场”这样的商业闭环。这类操作系统，我们不妨把它叫做现代操作系统。





## 操作系统的边界在哪里？

架构的第一步是需求分析。上一节我提到了在架构设计过程中，需求分析至少应该花费三分之一的精力。通过这一节我们对操作系统演进过程的回顾，你可能更容易体会到这一点。

当我们说要做一个操作系统的时候，实际上我们自己对这句话的理解也是非常模糊的。尤其是我们正准备去做的事情是一个新生事物时，我们对其理解往往更加粗浅。

在本专栏开篇词中我也提过，架构也关乎用户需求，作为架构师我们不只是要知道当前的用户需求是什么，我们还要预测需求未来可能的变化，预判什么会发生，而什么一定不会发生。

我们可以问一下自己：我是否能够预料到，有一天支付（Pay）系统会成为操作系统的核心子系统？如果不能，那么怎么才能做到？

操作系统的边界到底在哪里？

要回答这个问题，我们需要看清楚这样三个角色的关系：

硬件（个人计算机、手机或其他）；

操作系统；

浏览器。

首先我们来看操作系统与硬件的关系。如果操作系统厂商不做硬件会怎样？我们知道个人计算机（PC）市场就是如此。微软虽然占据了 PC 操作系统（DOS/Windows）绝大部分江山，但是它自身并不生产硬件。这里面，PC 兼容机的发展对 DOS/Windows 的发展有着至关重要的支撑意义。它让操作系统厂商有了独立的生存空间。

到了移动时代，Google 收购 Android 后，通过免费策略占领移动操作系统的大半江山，一定程度上复制了微软的过程，但实际上并没有那么理想。

首先，Android 是免费的，Google 并没有从中收取软件 License 费用，而是借助 Android 的市场占有率来推动 Google 的服务（例如搜索、Gmail 等等），通过 Google 服务来获取商业回报。

其次，iOS 操作系统引入的“账号 - 支付 - 应用市场”的收税模式，受益方是硬件（手机）厂商，而非操作系统厂商。其中最关键的一点，几乎所有手机厂商都不接受把支付（Pay）这个核心系统交给 Google。

最后，不止支付系统，一旦手机厂商长大立足，Google 服务也会被逐步替换。所以 Google 和 Android 手机厂商之间的联盟并不可靠，养肥的手机厂商会不断试探 Google 的底线，而 Google 也会尝试去收紧政策，双方在博弈中达到平衡。

之所以会这样，我觉得原因有这么几个：

其一，历史是不可复制的，人们对操作系统的重要性认知已经非常充分。所以大部分手机厂商，都不会放弃操作系统的核心子系统的主控权。Android 系统的开源策略无法完全达到预期的目标，这也是 Google 最终还是免不了要自己做手机的原因。

其二，手机是个性化产品，硬件上并没有 PC 那么标准化。所以个人计算机有兼容机，而手机并没有所谓的标准化硬件。

分析完操作系统和硬件的关系，我们再来看它和浏览器的关系。在 PC 时期，操作系统和浏览器看起来至少需求上是有差异化的：操作系统，是以管理本地软件和内容为主（对内）。浏览器，是以管理互联网内容为主（对外）。

但，这个边界必然会越来越模糊。

操作系统不涉足互联网内容？这是不可能的。应用市场（AppStore）其实就是典型的互联网内容；而另一方面，在浏览器的生态里也有一些特殊角色：网址导航、搜索引擎、Web 应用市场，它们共同构成了探索互联网世界的“地图”。

问题在于：

操作系统、浏览器和（互联网）搜索引擎的关系是什么；

移动时代的浏览器会是什么样的；它和操作系统的关系又如何相处？

欢迎把你对这几个问题的想法告诉我。

## 结语

让我们简单回顾下今天我所讲到的内容。

从客户需求来说，操作系统的核心价值在于：

**实现软件治理，让多个软件和谐共处；**

**提供基础的编程接口，降低软件开发难度。**



从商业价值来说，操作系统是刚性需求，核心的流量入口，兵家必争之地。所以，围绕它的核心能力，操作系统必然会不断演化出新的形态。

我们把引入了“账号 - 支付 - 应用市场”商业闭环的收税模式的操作系统，称为现代操作系统。

操作系统的边界到底在什么地方？我们通过分析硬件、操作系统、浏览器三者的关系，也做了定性的分析。这样的分析将有助于你对需求发展做出预判。

最后，你可以在留言区给我留言，分享你对于操作系统技术、商业的看法，让我们交流。



# 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 05 | 思考题解读：如何实现可自我迭代的计算机？

下一篇 07 | 软件运行机制及内存管理

精选留言 (56)

写留言



郭小菜

2019-05-03

👍 37

老师，请收下我的膝盖

展开 ∨



Jiayp

2019-05-03

👍 27

这才是从需求出发，以前学操作系统，上来就讲有哪些操作系统，操作系统的实现，从来没有考虑过，操作系统到底要实现哪些需求



coder

2019-05-05

👍 15

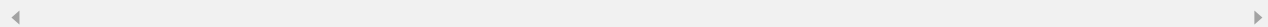
许老师，您好，虽然自己是科班出身，但是对于下面这个问题困扰了我很久：

1. 一般来说程序的运行需要OS的支撑，那么在BIOS之前，选择运行哪个操作系统那一段程序是怎么运行的？再问一句，编译器可以独立于操作系统运行吗？
2. 可以用C语言去实现很多其他的语言，比如说python, go等，那在C语言之前，这么一直追寻下去，会衍生鸡生蛋的问题。但是编程语言又具有自举的功能，那自举是怎么实...

展开 ∨

作者回复: 1、程序运行不需要操作系统支持，有bios支持就可以（把控制权交给它）。编译器可以独立于操作系统存在，而且它应该先于操作系统产生。

2、语言诞生的过程是这样的：机器码 =》汇编 =》C =》C写的汇编，C写的C（自举）。当然这个过程不需要每次新架构的cpu或操作系统都重新来一遍，因为人是聪明的，发明了交叉编译这样的东西，C =》新平台的C，这样就一下子完成整个语言的进化了。



公号-代码...

2019-05-04

👍 7

1 操作系统、浏览器和互联网搜索引擎的关系是什么？

随着技术的更迭，Web应用程序已经成为主流的软件形态，而浏览器是运行web应用程序的容器（传统的二进制应用程序也已被支持）；因此，浏览器已然成为PC时代的访问入口，从分层的技术架构上来看，操作系统已成为浏览器的底层，而浏览器已成为操作系统的入口，用户甚至无法感知操作系统的存在，所以从未来的发展趋势来看浏览器完全可...

展开 ∨





Enthusiasm



关于操作系统，浏览器，搜索引擎，可以归结为平台产品。我并不认为每个架构师都有机会做出一款平台产品，而且，我认为平台产品还是类似于基础服务，并不能带来很高的收益，获得收益关键是如何形成一个商业闭环，让钱乖乖流进来。操作系统已然形成不了技术壁垒，也就起不到商业壁垒的目的。未来能形成商业壁垒的，我认为是产品差异化，物联网时代想方设法做出新产品的机会还是蛮多的。微信小程序想让所有程序都集成到微...  
展开 ∨



好好学习

2019-05-04



k8s正在接管集群操作系统的角色，服务治理是它根本停不下来  
展开 ∨



苟范儿

2019-05-06



老师现在主要提到的是终端（PC/Mobile）类的操作系统。当前的云计算底层的系统，以及未来 IoT 边缘系统会有什么特殊之处？

作者回复: 挺好的问题。后面会在第三章服务端开发再次讨论操作系统，会探讨它的演进历程和方向。



Enthusiasm

2019-05-03



老师您的意思是让我们一上来就多考虑一下系统的“收费需求”或者“商业模式”。嗯，很好。总结一下各个公司的成功途径，就是先要满足大量刚需或者打破现有商业壁垒，然后靠技术、工艺和法律形成新的商业壁垒迅速崛起，最后依靠积累的用户基数和粘性，获得规模效应，开发新产品以获得更高的溢价等等。  
做硬件和做软件要相结合，只做上游服务并不能形成良好的商业闭环，所以IoT+人工智...  
展开 ∨



frostawn

2019-05-09



许老师，小白请教个很困惑的问题。  
为什么pc可以有兼容机，让操作系统windows收费或者linux免费各占半壁江山，但是手机

不能有兼容机呢，而说手机是个性化产品，不能标准化，他们再怎么个性，也脱离不了前面讲过的冯诺依曼体系吧。而且现在现实也是有安卓和ios两大阵营。所以并不是技术上没做到兼容，而是商业盈利模式的区别，ios可以利用支付闭环将流量变现，安卓拿不到支...  
展开 ▾

作者回复: 和技术没关系，是人性的问题。手机是人体器官的延伸，是门面，用户不会接受白牌机的，所以手机不是技术上做不到兼容机，而是人性上不会接受白牌机。



apk

2019-05-08

👍 3

对于“PC无法摆脱操作系统控制但是手机厂商可以摆脱Google的控制形成自己的商业闭环”这一点不是明白？

Pad、一体机这些能不能也形成自己的商业闭环呢？

展开 ▾

作者回复: 历史阶段不同。PC的历史再演一遍，可能会是完全不同的结果。



晓凉

2019-05-04

👍 3

隐隐感觉到架构的主要难点在于对需求的前瞻性判断，这要求的不仅仅是技术能力。目前几乎所有的架构课程，都是基于确定的需求来讲技术架构，例如秒杀系统怎么做高可用高并发。能听到许老师这么高视角讲架构，实在幸运。

展开 ▾

作者回复: 架构在于创造，如果你从事的事情总是重复别人，那这个公司又有何价值？即使有所参考，也应该有自己的精气神，这个精气神是需要架构师把它干出来的



冰块

2019-05-03

👍 3

操作系统会因为浏览器退出历史舞台吗

展开 ▾

作者回复: 不会



**Smallfly**

2019-06-06

👍 2

许老师，我今天重新回顾了一遍这篇文章，以需求为出发点的写法，给读者的印象更加深刻，也更好理解，非常赞。但是我有两个问题想请教：

1、

对于六大系统的划分，文件和内存被统一为存储管理，为什么不把输入输出合并为设备...  
展开 ∨

作者回复: 1、这里的输入与输出设备，是指用户输入和面向用户交互的输出设备，不同于冯诺依曼结构中提到的输入输出设备（泛指所有设备，包括存储设备和网络设备）。输入与输出设备管理合并为一章并不是因为它的内容少，而是因为它不是本章的重点。

2、PC 兼容机是指白牌机，没有品牌或品牌几乎没有知名度。而且很多白牌机连操作系统都没有预装，硬件厂商没有意识到预装是一个巨大的市场。



**Cordova**

2019-05-09

👍 2

根据老师的理论过程来看，个人觉得操作系统它是一个没有明确边界的，只有放在历史（时间）的维度上它才有边界的表现，其实也不需要去深究其边界、在pc时代互联网的出现的确让人感觉浏览器赋予了不同反响的历史使命，以后将发挥更大的价值，移动时代发现我们所需要的内容不仅仅可以在浏览器上才能获取到，在不同的内容平台也能获取到自己想要的内容，而现在正是如此、要说操作系统的下一次演进、我想只有两个方向：更...  
展开 ∨

作者回复: 挺好的思考



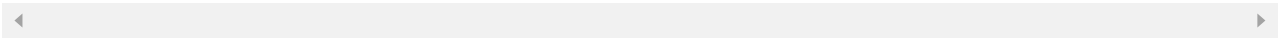
**gbfa**

2019-05-06

👍 2

这种对需求的前瞻性探索挺重要，但同时感觉也是最难的，应该如何培养呢 ~  
展开 ∨

作者回复: 很多时候是思维方式的转变。首先要尝试去做前瞻, 预测错了并不可怕, 但可以事后复盘到底是缺失了什么重要的信息让你判断出现了偏差。



**zjh**

2019-05-05

👍 2

感觉不论操作系统还是浏览器, 一旦成熟就会变为基础设施层, 人们就默认它的存在, 从而就会出现建立在其之上的应用来取代他们成为新的入口



**涵**

2019-05-04

👍 2

大胆设想, 未来对大众消费者来说会持有各种终端, 仅是终端而已, 专注于输入输出, 绝大多数计算和存储能力由云服务完成。终端设备无需复杂操作系统。设备开启接入网络, 可能会在云端有一个针对这种设备这个账号的操门户, 用户通过这个门户实现与各种应用的交互。因此, 可以预见, 云端上对于用户设备门户服务的争夺会很关键。微信小程序就有这个意思。这个门户服务就像是用户终端设备的操作系统, 但是部署在云端, 对用户...

展开 ▾



**Geek\_f26a6...**

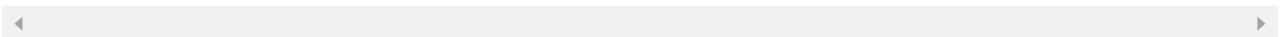
2019-05-03

👍 2

老师写的不错, 例子也恰当, 只是花的篇幅有点儿多。

展开 ▾

作者回复: 我觉得在计算机的基础架构体系中, 操作系统是最值得研究的, 最值得反思和复盘的。本章从架构的第一步需求分析入手, 我们花费的主要的篇幅集中在需求的演化上。



**Smallfly**

2019-06-06

👍 1

浏览器只是操作系统之上的一个应用程序, 相比其他应用程序, 它主要用于互联网内容的展示, 重度依赖于操作系统的网络管理子系统。

浏览器方便了不同内容之间的跳转, 至于搜索其实并不算浏览器特有的功能, 用命令行也可以进行互联网搜索。...



展开 ∨

---



**Y**

2019-05-07

👍 1

第一次从商业模式，盈利上理解为什么要这样做架构~~~

展开 ∨