

“全场景通用”操作系统解决方案的发展道路研究

刘轩铭，3180106071

（浙江大学，计算机科学与技术学院，软件工程专业）

前言：操作系统提供一个让用户与系统交互的操作界面。未来是万物互联的时代，操作系统如何实现不同终端之间的数据共享和应用对于不同设备终端的适配问题，是一个重要的难题。为了解决这一问题，微软在 2015 年提出了“UWP”的解决方案，通过 API 的调用和 UI 方面的算法，解决了应用适配的问题。但是由于用户生态的问题和技术本身的局限性，该方案逐渐淡出人们的视野；2020 年，华为发布了“面向全场景”的操作系统“鸿蒙”。通过使用分布式技术，“鸿蒙”系统解决了数据共享，多平台互联的问题。该难题有了较好的解决方案。在文章的最后，有作者关于本文研究过程中提出的几个问题，和作者自己的思考。

一、操作系统全场景通用使用难题

操作系统是管理计算机硬件与软件资源的计算机程序。操作系统需要处理如管理与配置内存、决定系统资源供需的优先次序、控制输入设备与输出设备、操作网络与管理文件系统等基本事务。同时，操作系统也提供一个让用户与系统交互的操作界面。

操作系统可以被应用在 PC 终端，手机，手表以及冰箱等嵌入式操作系统中。由于不同操作系统的系统服务层和应用层所采用的技术和原理不同，一个应用程序在不同终端使用时的效果有很大的差别。以上所说的“不同”，主要体现在应用层呈现界面（即 UI）的不同以及数据上的不共享、不共容。

以较为常见的使用场景为例：日常使用的手机、电视、笔记本以及冰箱洗衣机、扫地机、净水器、汽车、灯光等家具，这些产品的嵌入式操作系统核心硬件架构都不一样，电脑可能是 X86 架构，手机可能是使用的尖端 ARM 技术，电视是中高端 ARM 技术。在这些不同硬件架构的设备上部署应用，有着很大应用体验上的不同，例如目前就无法在冰箱和手机上共用一个京东账号进行操作，同时两者对用户的接口也有较大区别。

以上的问题导致，同一个产品开发商，对于不同平台使用的产品，需要开发多套代码进行适配，而其中大部分工作其实都是重复工作，从而降低了开发效率。所以，新的技术需要解决对于 UI 表现层的适配问题；进一步而言，还需要解决多端数据共

享的问题。

二、微软的“UWP”解决方案

1. 什么是“UWP”解决方案

为了解决上述问题中，对于不同硬件产品，用户体验一致性差的问题，以及一套代码对于不同硬件产品的前端兼容问题，微软为 Windows 提出了 UWP（Universal Windows Platform）解决方案——Windows10 通用应用程序。

UWP 应用程序，其最大的特性是它可以实现真正的“全设备通用”，也就是说，它并不是为某一个确切的设备终端所设计的，而是面向于一系列相同的操作系统平台。2015 年，微软在 Windows10 的发行版中首次使用了 UWP 技术。开发者只需要编写一次 UWP 程序的代码，就可以在所有的 Win10 设备上运行。无论是电脑、手机、平板电脑、Surface Hub 大屏设备，还是基于 Win10 的全息眼镜 Hololens，都能够完美地运行起来。这不仅为 UWP 的开发者们提高了工作效率，还为普通用户们提供了前所未有的便利。

2. “UWP”的设计思想和实现：API 特化和 UI 自适应算法

操作系统中有 API 的概念。API 是带有版本号和标识符的最底层系统调用的集合。在 Windows 中，它们大部分是对于底层系统调用的封装。它又分为

通用 API 和各个子系统 API 等。

通用 API 相对而言比较特殊，它不是任何操作系统的基础；相反，通用 API 是各种操作系统的功能能够正常呈现到每种设备中的基础。

每个子系统 API 族在通用 API 基础上添加的适配于各种设备的自有的 API。使用 API 族的优点在于开发一次，可以到处运行。无论用户使用哪种设备，手机，平板或 PC，API 可以自适应于底层的设备。从而 APP 可使用自适应代码实现动态获取设备特征，适应设备。

在开发中，开发者只需要给工程添加包含特定子 API 族的 SDK 文件，就可以完成对于特定设备的自适应。



图 2-1 不同设备的尺寸自适应难题

同时在 UI 设计上，Windows 设计了自适应的控件和输入环境，这主要依赖于像素的自适应算法。UI 元素通过调整其布局 and 比例响应应用所运行屏幕的尺寸。主要是在像素上，UWP 可以自动调整 UI 元素的大小，以使它们在所有设备和屏幕大小上清晰可见并易于交互。

当应用在设备上运行时，系统将使用算法来使 UI 元素在屏幕上的显示方式规范化。此缩放算法考虑了观看距离和屏幕密度（每英寸像素），以针对感知大小（而不是物理大小）进行优化。该缩放算法确保用户可从 10 英尺远处识别 Surface Hub 上高 24 像素的字体，和从几英寸远处识别 5 英寸手机上高 24 像素的字体一样。

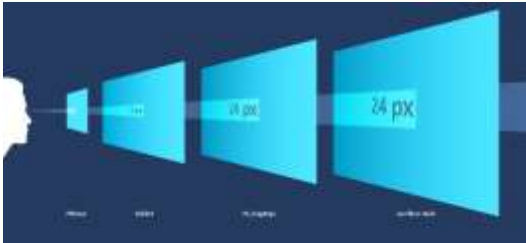


图 2-2 UI 对于像素的调整示意图

3. “UWP”设计的应用效果：效果良好但应用环境较

差，具有较大局限性

由于调用子 API 族的行为经常发生，所以实际上 UWP 应用是完全受操作系统控制的，同时存放数据也只会存放环境变量%APPDATA%和%TEMP%等系统规定的目录里。因此 UWP 应用是不会对你的系统造成任何伤害的，在卸载时也能保证完全卸载干净。

相较而言，传统应用在运行时有时要求以管理员身份运行，而这样的做法是较为危险的。

UWP 设计被广泛应用在了多种基于 Windows10 操作系统的设备上。目前大量的应用程序是应用这一思想进行开发的。所以，我们可以看到在 Windows 上大量的应用都有良好的自适应效果。



图 2-3 bilibili (APP) 竖屏展示图



图 2-4 bilibili (APP) 横屏展示图

在上面的例子中，UWP 的思路是这样的：如果的窗体被限制窄了，不能展示更多内容，就触发竖屏模式。如果窗体被拉宽了，就触发横屏模式，此时多出来的展示区域就被用来显示更丰富的内容。

然而，由于本来最应该搭载这种设计思想的 Windows Phone 设备一直销量不佳，在市场份额上下滑非常严重；同时，微软的智能家居，车载操作系统等更多场景的开发和应用并不完备。所以这一设计被局限在了以 Windows10 PC 为主的少量设备上，并没有得到广泛应用。

此外，UWP 还有一些局限性。由于操作系统对 UWP 管控较严，UWP 应用的速度会比传统应用低；、因为各种限制，UWP 应用的开发比普通应用困难，因此许多开发人员都对开发 UWP 应用不积极。

最后，UWP 并没有解决数据共享的问题。他仅仅是同一块 APP 在应用层上适配不同设备的一种解决方案。

三、 鸿蒙：“面向全场景”的操作系统

1. 鸿蒙使用分布式的方案解决全场景通用难题

2020 年 9 月 10 日，华为在 HDC 开发者大会上如约发布鸿蒙 2.0，并面向应用开发者发布 Beta 版本。公司给出的口号是：这是一款“面向未来、面向全场景”的分布式操作系统。



图 3-1 “鸿蒙”超级终端万物互联示意图

鸿蒙本身是一个分布式的跨硬件平台的物联网系统。通过分布式的操作，它解决了数据共享的问题。这意味着用户可以使用华为的产品，如 PC 终端来控制华为的智能冰箱等其他产品。

此外，鸿蒙系统允许多终端开发 IDE，多语言统一编译，并利用分布式架构 Kit 提供屏幕布局控件以及交互的自动适配，支持控件拖拽。从而，开发者可以基于同一工程高效构建多端自动运行 App，实现真正的一次开发，多端部署，在跨设备之间实现共享生态。

以上便是鸿蒙“面向全场景”的含义：既解决 APP 多端适配的问题，又解决了终端之间数据共享的问题。

2. 鸿蒙的实现方法和架构

鸿蒙系统整体上分为内核层，系统服务层，应

用框架层和应用层四个模块。其中在系统服务层，系统采用了分布式的架构，具体如下图。



图 3-2 “鸿蒙”技术架构图

具体而言，鸿蒙系统有着如下的技术特性，使得全场景通用成为可能：

- 鸿蒙使用了“分布式软总线技术。分布式软总线是多种终端设备的统一基座，为设备之间的互联互通提供了统一的分布式通信能力，能够快速发现并连接设备，高效地分发任务和传输数据。通过公共通信平台，在分布式数据管理，分布式能力调度和虚拟外设支持下，系统允许开发者在类似同一终端下一样开发跨终端的分布式应用。这实现了跨终端无缝协同体验。
- 鸿蒙系统使用了“确定时延引擎”和“高性能 IPC 技术”，使得系统天生流畅。“确定时延引擎”保障优先级高的任务资源将优先调度，应用响应时延降低 25.7%；而微内核的结构小巧使 IPC（进程间通信）性能大大提高，进程通信效率较现有系统提升 5 倍。这也使得数据共享成为可能。
- 鸿蒙系统的开发时通过统一的 IDE 支撑。实现了一次开发，多端部署，实现跨终端生态共享。华为方舟编译器是首个取代 Android 虚拟机模式的静态编译器，可供开发者在开发环境中一次性将高级语言编译为机器码。此外，方舟编译器未来将支持多语言统一编译，可大幅提高开发效率。

华为的鸿蒙系统就是基于以上的技术，进行了“全场景通用”的操作系统实现。

然而，以上的技术是否真正的成功和成熟，还未能验证。据报道，全新搭载“鸿蒙系统”的华为手机将于 2021 年上半年上市，或许，那时便是我们

见证华为解决“全场景通用”这一难题的时刻。

么相信它会在未来的生态下不断迭代，变得更好。

四、 关于这一问题的思考

1. 为什么微软的“UWP”方案难以成功？

首先，“UWP”技术本身存在缺陷。尽管它较好的解决了应用程序在不同类型设备上的适配问题，但是也存在诸多问题：例如引用运行速度较慢等。

其次，微软的 Windows 应用缺乏一个较好的用户生态。其实由于 Windows Phone 的用户生态较差，用户较少等问题，缺乏开发者来开发足够数量的 UWP 应用。同时，在微软发布 Windows 10 之后的迭代版本中，也缺乏对于 UWP 生态系统的维护和支持。这使得 UWP 逐渐淡出人们的视野。

2. 相对于“UWP”方案，“鸿蒙方案”有着怎样的优势？

首先，鸿蒙方案并不只是局限在 UI 上的适配这一问题。其实要解决 UI 适配问题，有很多的方案都可行。例如，腾讯开发的微信小程序就能够实现在不同设备上的微信应用中挂在小程序。鸿蒙系统给出的更优解是：通过分布式数据共享，使得应用在不同终端上能够万物互联。

例如，戴森（Dyson）的台灯，拥有开一个叫 Link 的工具。该工具可以操控戴森台灯，但是却不能操控同样是戴森品牌的吸尘器。因为产品形态和用途的显著不同，很多公司都没有能力为旗下的产品开发一个共同的应用入口和应用平台，这也就给了目前华为的一个历史性的机会。

鸿蒙系统会解决这个问题，支持与鸿蒙系统互通的手机（目前看是华为，以后也不一定只有华为手机能互通鸿蒙），不需要装这些品牌商的 APP，就可以直接对它们的家电产品（同样装载了鸿蒙系统）进行操控。

此外，由于华为目前在国内手机市场拥有着最大的市场份额，也是世界范围内拥有手机市场份额最大的国产品牌，华为有着令人信赖的用户生态。在目前国内外贸易局势紧张的氛围下，也会有更多有热情的开发者主动进入华为的“鸿蒙生态”，一起加入开发的道路中。

如果我们的系统真的能够达到预期的要求，那

刘轩铭：《“全场景通用”操作系统解决方案的发展道路研究》