

第 1 章

鸿蒙系统简介

要了解一件新事物，需要去追溯它的过去，去畅想它的未来。本书的主角是鸿蒙操作系统，我们将从多方面来阐述对这个系统的热爱以及对未来的信心。

1.1 智能手机操作系统

1.1.1 智能手机系统发展历史

时代变迁，新技术替代旧技术是技术发展的趋势，我们所处的时代，技术正在发生着翻天覆地的变化。新能源正在取代传统能源、电动汽车正在取代传统燃油车、5G 网络正在取代 4G 网络、新一代操作系统正在取代智能手机操作系统。这里的新一代操作系统，指的其实就是鸿蒙这样的万物互联系统。我们以智能手机操作系统的发展为例，慢慢引入对鸿蒙系统的讨论。

2017 年前，以诺基亚为代表的功能手机还是全球手机的领导者，经典的板砖造型和开机铃声，让人记忆犹新。当时的操作系统还是 Series30/40 等功能机操作系统。

2017 年 1 月，iPhone 横空出世，由于诺基亚的应用程序比不上 iPhone 的应用程序，加上诺基亚的固执，让其迅速被时代抛弃，全球市场份额快速衰减。

2018 年 9 月，Android 1.0 发布，此后的两年，Google 迅速发布了多个 Android 版本。智能手机操作系统以迅雷不及掩耳之势被 iOS 和 Android 两大系统占据。

在智能手机的竞争中，微软发力较晚，直到 2010 年，微软才正式发布了 Windows Phone 7.0 系统。比 iPhone 和 Android 足足落后了两年多，在这两年中，iPhone 和 Android 已日渐羽翼丰满。毫无疑问，相比 Android 和 iOS，Windows Phone 是一个非常尴尬的存在。与 Android 相比，Windows Phone 的闭源路线无法战胜 Android 系统的开源路线。与 iOS 相比，在友好甚至惊艳的用户体验上，Windows Phone 从来不是 iOS 的对手。再加上一个最为致命的点，就是生态，数以千万计的研发人员，不愿为了转入 Windows Phone 系统而熟悉一种新的开发方式。

2011年初，诺基亚和微软同时觉察到了危机，两大巨头宣布合作，诺基亚官方宣布放弃塞班（Symbian）品牌，转而投入微软的怀抱，全面支持 Windows Phone 系统，不过为时已晚。Android 和 iOS 的生态已经发展起来，想要颠覆他们的生态系统，实在太难。多年后，2019年12月，微软表示将暂停 Windows 10 Mobile 的更新，这意味着微软将放弃移动操作系统。作为份额仅占 0.1% 全球第三的手机操作系统 Windows Mobile 将彻底地退出历史舞台，英雄迟暮。

与此同时，2019年8月9日，华为在东莞举行华为开发者大会，正式发布鸿蒙操作系统，鸿蒙时代开始慢慢地含苞待放。在介绍鸿蒙系统之前，我们来研究一下 Android 和 iOS 操作系统走过的路。

1.1.2 智能手机的开放与封闭之争

在智能手机操作系统发展的过程中，特别是 Android、iOS、Windows Phone 的竞争过程中，在面临的所有问题中，操作系统的开放与封闭问题非常值得思考。

开放与封闭本身是一对反义词，同时，这也代表着当今世界智能设备中举足轻重的两款操作系统的态度。其中，Android 选择的是开放，iOS 选择的是封闭。由于这两个系统几乎是同时跨入智能手机这个风口的，无论是开放还是封闭，这两个操作系统都展现出了巨大的商业价值，目前其市场价值不分伯仲。

1. 开放战略

开放即开放源代码，开放生态合作伙伴，开放应用商店等，Google 即选择的开放这条路。开放具有如下一些优势。

- 商业风险低。开源软件的生命周期长，维护者多，不用担心开源软件的 bug 没人修改，如果闭源，会担心闭源软件的公司倒闭。开源系统如果 bug 实在没人修改，那么自己修改即可。以 Android 为例，维护者有众多国际一线公司，首先是 Google 自身，其次有华为、高通、小米、Sony 等。对于 Android 来说，基本是大型的公司共享代码，对于个人贡献者来说，相对较少，原因是担心出了问题，找不到个人贡献者来修改。由于基本是国内外顶尖科技公司在贡献代码，所以其代码质量非常高。
- 开源软件能保证好的品质。相对于大多数闭源产品，开源社区一般有众多免费的设计、编码、测试、维护人员，他们以满腔热血在工作，不计报酬。因为同一份代码被多个人阅读审查，所以一旦发现 bug，整个社区也会积极修复。
- 相对安全性更高。开源软件中一般很少有病毒，因为不会有人把病毒放在开源软件中，让成百上千的研发人员发现。同时，由于参与研发的人员多，开源软件能够更快

地发现软件的安全性 bug。

- 成本低。开源社区集合了大量的软件开发者，很多开发者是不需要工资的，开源工作者都是默默地付出劳动成果，为理想而战。

2. 封闭战略

封闭一般指封闭源码，即封闭部分接口和权限。例如，2014 年前的 iOS 系统，是不允许安装第三方输入法的，对于在桌面计算机上使用习惯了讯飞、搜狗输入法的人来说，iOS 自带的输入法使用起来确实不便，这就是封闭的坏处。但是封闭也是有好处的，最近国内某几款输入法被苹果商店下架，就是因为其收集了大量用户文字输入信息，用于分析用户画像，提供给第三方电商平台，导致很多用户的数据被泄露，这就是开放导致的一些问题。例如，我们经常输入一些贷款、抵押的文字，就很快会有人打电话过来问我们是否需要资金，这就很可能是我们的信息被泄露了。

封闭具有如下一些优势。

- 闭源有助于增强品牌优势。一直以来，iOS 系统是较为封闭的。iOS 系统只能用于自身的硬件中，其他厂商不能使用。这种封闭性让 iOS 有更一致的体验，很长一段时间，iOS 的流畅度远远高于 Android 系统，良好的用户体验对于树立品牌形象有很大的帮助。事实上，苹果的 iOS 系统一直以来都是非常封闭的，除了 iPhone、Mac、iPad 等自家衍生的产品，没有其他品牌使用，这也就保持了苹果自家 iOS 系统的封闭性，因此才能够做到完全适配自家的硬件设备，最终让用户得到更加流畅的体验，这才是 iPhone 手机的竞争力所在，自然不能够开源。
- 满足核心需求。苹果不在乎满足所有用户的需求，它只是找到一个卖点，聚集用户打造基础生态圈，再通过基础生态圈形成辐射影响，用户的体验和习惯是可以被影响和引导的。
- 安全性更高。众所周知，iOS 的病毒是少于 Android 的。这是因为 iOS 的应用是通过苹果商店审核的应用，苹果的审核很严格，要求很高，恶意或病毒程序，审核通不过，所以病毒比较少。

1.2 鸿蒙系统发展历史

在华为“2012 诺亚方舟实验室”专家座谈会上，任正非提出要做终端操作系统，以防患于未然，要在“断了我们粮食的时候，备份系统要能用得上”，而这就是“鸿蒙”操作系统的起点。鸿蒙系统主要经历了如下几个发展阶段。

- 2012 年，华为开启了操作系统“B 计划”，开始规划自有操作系统“鸿蒙”，不被卡脖子是华为的重要战略。
- 2016 年 5 月，华为消费者 BG 软件部开始立项研发“分布式操作系统 1.0 版本”，这就是鸿蒙系统的雏形。
- 2017 年 5 月，华为消费者 BG 软件部研发完成“分布式操作系统 1.0 版本”，开始研发 2.0 版本。
- 2018 年初，华为消费者 BG 业务给任正非做汇报，任正非非常认可自主研发的操作系统。
- 2018 年 5 月，华为消费者 BG 投资委员会投票同意“分布式操作系统”的研发，“分布式操作系统”正式成为 BG 核心项目。
- 2019 年 5 月，经 3 年研发，“分布式操作系统”已开始成熟，为了更好地推广，系统更名为“鸿蒙”。鸿蒙，是一个汉语词语，亦作“鸿蒙”。传说在盘古开天辟地之前，世界是一团混沌状，因此把那个时代称作鸿蒙时代，后来该词也常被用来泛指远古时代，鸿蒙系统寓意着一个新的伟大的物联网系统开始了。
- 2019 年 8 月，鸿蒙 1.0 系统发布，作为首批试点，荣耀智慧电视中搭载了该系统。同时，余承东表示，鸿蒙操作系统实行开源战略。
- 2020 年 9 月，鸿蒙 2.0 系统发布，该系统逐步开源，电视、手机、车机等设备可以使用鸿蒙系统。在小型设备领域，鸿蒙系统支持 128KB 到 128MB 的设备，并首先对这些设备进行了开源。
- 2020 年 12 月，鸿蒙系统面向移动开发者，推出鸿蒙 Beta 版。
- 2021 年 4 月，鸿蒙系统扩大开源范围，向内存 128MB 到 4GB 的设备开源，华为 Mate X2 手机开始使用鸿蒙系统。
- 2021 年 10 月，计划向 4GB 内存以上的设备开源，手机厂商可以修改系统源码以适应自己手机的特殊需求，这和 Android 走的是同一条线路。

目前为止鸿蒙系统已经发布了两个版本，分别是鸿蒙系统 1.0 和鸿蒙系统 2.0。下面对这两个版本做简要的介绍。

1.2.1 鸿蒙系统 1.0 介绍

2019 年 8 月 9 日，华为在东莞松山湖举行了一年一度的开发者大会，正式发布操作系统鸿蒙系统 1.0，发布会上宣布，荣耀智慧屏、荣耀智慧屏 Pro 都搭载了鸿蒙操作系统。

鸿蒙操作系统是一款全场景分布式操作系统，可按需扩展，实现更广泛的系统安全，主要用于

物联网，特点是低时延。分布式操作系统是一种面向未来的操作系统，华为定义的分布式操作系统具有如下特性。

- 多终端之间的能力共享，互为外设。例如，A 终端能操作 B 终端的摄像头，鸿蒙的前辈们是不具备这个激动人心的功能的。
- 系统与硬件解耦，弹性部署。鸿蒙系统支持 128KB 到 4GB 的设备，从微型设备到大型设备都支持。可以根据资源情况定制内核，增加或删除一些模块。鸿蒙系统支持低资源设备，这完全符合物联网设备的特性。
- 应用一次开发，多端部署。鸿蒙实现了应用跨终端运行，一次开发，可以将同一程序部署到不同运行鸿蒙系统的终端上，如手机、电视、车载系统中。

鸿蒙系统 1.0 实现模块化解耦，对应不同设备可弹性部署，弹性部署是指，可以根据硬件的情况，对系统进行裁剪，最小化部署到硬件设备上。鸿蒙系统 1.0 就已经构建了完整的四层架构，第一层是内核，第二层是服务层，第三层是开发框架层，第四层是应用层，为后续鸿蒙系统 2.0 的开发打下了良好的基础。

1.2.2 鸿蒙系统 2.0 介绍

2020 年 9 月 10 日，鸿蒙系统发布了 2.0 版本。鸿蒙系统 2.0 重点在分布式架构上做了大量的升级优化。在分布式软总线、分布式数据管理、分布式安全等方面进行了重大的更新，开发者能够从这些更新中获得更多的开发能力。在鸿蒙系统 2.0 推出的同时，华为也联合了国内众多知名的厂商，如美的、九阳、老板等智能家居公司，在这些品牌的智能家居上搭载鸿蒙系统，以后大家购买这些电器，就能够很方便地使用这些电器的功能了。例如，可以在冰箱的屏幕中，看手机里视频软件的视频，完全实现分布式的视频通信。可以通过鸿蒙系统，便捷地操作空调，任何设备（如空调）都可以看成是手机的虚拟设备。

1.2.3 鸿蒙系统与物联网

鸿蒙系统是新一代的物联网操作系统，这是其最大的特点。鸿蒙系统从立项之初就不是只为手机而生的，而是为了物联网。从这个角度分析，鸿蒙系统并不是为了取代 Android 和 iOS 系统，而是一个全新的物联网系统。为什么物联网现在需要一个全新并统一的操作系统呢？这要从物联网说起。

1. 物联网的两层含义

从物联网的概念中，我们能够发现物联网有两层含义：第一，物联网仍然是基于互联网

的，是使用互联网的基础设施，是互联网的扩展；第二，物联网的创新是将人与人的连接，扩展到人与物、物与物的连接，任何物之间都可以进行信息交换，即所谓的万物互联。

传统互联网中的信息交换一般要通过 OSI 参考模型（七层网络模型）来处理，物联网则不一样。物联网的通信可能会通过多种设备，如蓝牙、红外、ZigBee、光通信、RFID 等技术，采用的协议栈多种多样，设备与设备间互联非常困难，需要互相兼容，才可能通信，所以对操作系统的要求非常高。而目前没有这样的操作系统，鸿蒙最大的优势为其轻量化的内核，能容易地支持各种协议、各种设备，让各种协议、设备高效地互联起来。

2. 物联网时代鸿蒙的巨大优势

操作系统黑暗森林法则是赢者通吃、占据市场，生态基本无法颠覆。操作系统无法通过复制商业模式，投资巨额资金成功。操作系统必须在市场萌芽期快速占据市场，才能够立于不败之地。这个法则，桌面时代微软的 Windows 如此，智能手机时代 Android、iOS 系统也如此。所以摆在鸿蒙系统面前的主要难题是生态问题，不能和 Android、iOS 抢生态，因为突破 Android、iOS 的生态异常困难，只能去打造一个新的生态，即物联网操作系统生态。

目前，智能手机操作系统已经发展到了成熟期，新意越来越少，操作系统的风口正在从智能手机操作系统转向物联网操作系统。在物联网上，相比 Android、iOS，鸿蒙系统是做得最好的，原因是 Android、iOS 本身并不是为物联网设计的操作系统，而鸿蒙从诞生开始，就是一个物联网操作系统。

软件银行集团董事长孙正义在演讲中说，物联网将会引领下一轮技术爆炸，就像历史上寒武纪爆发形成了无数新物种一样，到 2035 年，联网的物联网设备数量就将达到 1 万亿，届时，平均每个人会有 100 个设备联网。

这些设备的每个个体都和手机有很大的差异，它们的构成、标准化各不相同，这就需要一个新的操作系统支撑如此异构的硬件设备。目前，鸿蒙正是这样一个系统。

鸿蒙系统不是以往操作系统的一种重复，而是全新的一种操作系统。如果顺利，属于鸿蒙系统的物联网时代即将到来。

1.3 鸿蒙系统的特点

计算机、手机、平板电脑、物联网设备的操作系统生态一般由硬件、操作系统、应用程序组成，如图 1-1 所示。

操作系统可以控制和管理计算机硬件和软件资源，合理地各类作业进行调度，以方便用户使用程序。操作系统对下必须要拥有控制硬件的能力，对上为应用程序提供底层支撑，为应

用提供访问硬件（如内存、磁盘、鼠标、键盘、相机等各种外设）的能力。



图 1-1 操作系统生态

从系统设计来看，一个操作系统的体系架构一般分为用户态和内核态。内核态中运行着系统内核。内核处于一个操作系统最核心的地位，其本质上是一种系统软件，主要控制计算机系统的硬件资源，为应用软件提供运行的支撑环境，如图 1-2 所示。

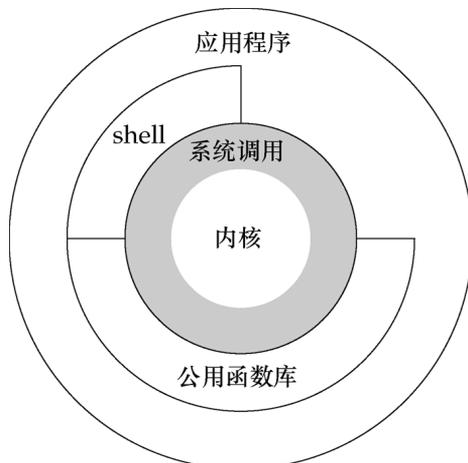


图 1-2 操作系统体系架构

内核向外提供功能，是通过 API 函数提供的，术语叫作系统调用。系统调用是操作系统提供功能的最小功能单位，系统调用函数一般有：

- 进程管理函数；
- 文件管理函数；
- 系统控制函数；
- 内存管理函数；
- 网络管理函数；
- 用户管理函数。

系统调用函数高度抽象、粒度非常细，一般一个操作系统中的系统调用函数较为稳定，不会经常变化函数签名。系统调用函数的个数也非常少，如 Linux 一般只有 250 个左右、

FreeBSD 有 320 个左右。基于 Linux，Android 系统调用和 Linux 几乎一致。

系统调用都属于细粒度的操作函数，要实现一些复杂的功能，就需要组合多个系统调用来实现，这就形成了一些公共函数库。

此外，在图形界面还未流行前，出现了一种 shell 命令程序，用户主要通过 shell 来和系统交互。

了解了内核的一些功能，下面我们主要对内核的分类做一些介绍，同时引出鸿蒙系统使用的是什么类型的内核。

1.3.1 内核特点简介

一个操作系统最核心的部分是其内核（kernel），内核就是操作系统的最主要代码，我们也可以简单地认为内核就是指操作系统。内核分为微内核和宏内核。下面对两种内核进行简要的说明。

1. 什么是宏内核

宏内核，又称单内核，是一种传统的内核结构。宏内核定义所有核心的程序都是在内核空间运行的，宏内核中的所有服务都是以特权模式执行。宏内核要求所有服务模块都在同一个地址空间运行，所有服务使用共享的地址空间、内存空间，如果某个服务模块出现了 bug，错误地操作了其他服务模块的地址空间，就会导致其他模块运行错误，从而使整个系统出现问题。因为很多服务都会运行在宏内核中，模块之间不能互相干扰，所以对模块的设计、开发、功能，性能上都有很高的要求。

宏内核从设计之初，就重点考虑了效率问题，宏内核模块由于使用的是同一个地址空间，所以它们之间能够更直接地通信，通信效率更高，从而使得运行效率也会更高。

但是宏内核也有天生的缺陷，其稳定性相对较差，由于各个模块运行在同一地址空间，如果某个模块出现 bug，那么会导致整个系统出现异常。所以，宏内核如果设计及开发的质量不佳，更容易导致稳定性差的问题出现。

UNIX、Linux、iOS 系统是宏内核。

2. 什么是微内核

微内核是提供操作系统核心功能的内核精简版本，它的核心设计原则是：能多小就多小。

微内核提供一组“最基本”的服务，如进程调度、进程间通信、存储管理、I/O 设备管理。其他服务，如文件管理、网络服务等并不是一个微内核必须有的，例如，一个蓝牙设备，就不需要网络服务，仅仅只需要文件管理服务。

微内核除了“最基本”的服务外，在有需求的情况下，怎么支持其他复杂的服务呢？这时候可以通过接口的方式，其他服务模块通过接口和微内核交流，从而扩展系统的功能。

Windows、鸿蒙是微内核。

3. 鸿蒙系统微内核的优势

鸿蒙使用的是微内核，Android 系统采用的 Linux 内核，Linux 内核属于宏内核，所以 Android 属于宏内核，iOS 也是宏内核。宏内核将所有的模块放在一个内核中，如文件系统、内存管理、网络管理、磁盘管理、设备驱动等。其中设备驱动是内核代码量最大的模块，为了支持众多系统，一般编译的时候，会将常用驱动编译到内核中，从而让内核镜像更大。

随着操作系统的功能越来越多，内核也越来越庞大，宏内核面临的挑战凸显出来，主要有两方面：

- 宏内核操作系统代码量非常庞大。Linux 3.3 的内核约 1500 万行，且每年再以 100 万行的速度增长。虽然全球有约 8 万人在参与 Linux 内核的开发和测试，但是如此大的代码量及系统复杂度，Linux 内核潜在的问题也相当多，并且修改起来非常复杂。
- 大量的驱动程序、模块集中在一个内核中，导致内核模块之间的关联性强，可扩展性差。内核之间的模块互相依赖，导致在修改某个模块时，可能会影响到其他模块。在多硬件适配的工作中，某些服务要匹配不同的硬件，要做大量的适配工作，导致服务不稳定，经常变化。这一点在物联网硬件中表现得尤为突出，物联网设备种类繁多，变化及组合方式多样，内核要适应这些设备，需要针对不同的物联网设备做大量的适配工作，这也是物联网宣传了很多年，但是目前流行程度还是相对较低的原因之一。

鸿蒙微内核完全克服了宏内核的这些缺点，将内核做到足够精简。内核只保留关键的一些系统服务，其他服务在内核之外的用户态去执行。微内核中只保留核心的进程管理、内存管理、进程间通信，其他的非核心服务放到非内核中去执行。如文件系统、网络协议、一些驱动程序等。这样做的好处如下。

- 高扩展。由于很多服务不在内核态，服务开发者可以自行开发、裁剪服务，并直接在用户态运行服务，对内核不会有任何影响，通过这种方式实现了高扩展。
- 高可靠性。微内核的代码量非常少，可靠性高。用户态的服务不会影响核心态的服务，从而实现高可靠性。
- 安全性高。微内核由于只实现了核心功能，代码量非常少，可以通过多次 review，不断地进行黑白盒安全测试，证明内核的安全性。
- 可维护性高。内核态运行稳定，基本不需要维护。运行在用户态的服务，可以正常启动、停止，即使出现问题，也不会导致内核崩溃。

1.3.2 鸿蒙系统分布式技术特性

除了鸿蒙微内核的这些优势，鸿蒙系统还解决了很多实际场景中的痛点，可以将软件与软件、软件与硬件、软件与资源快速共享。为了实现这些场景，鸿蒙系统引入了如下一些关键技术：

- 分布式软总线；
- 分布式设备虚拟化；
- 分布式数据管理；
- 分布式任务调度。

下面对这几点进行详细的阐述。

1. 分布式软总线

分布式软总线是解决物联网硬件设备之间无缝通信的总线架构，类似于企业服务总线（ESB）。分布式软总线为设备之间的互联互通提供了统一的分布式通信能力，各种设备，如手机、平板电脑、智能穿戴、智慧屏、车机、智慧家居等都能通过注册到分布式软总线，进行快速的数据、命令交互。分布式软总线为设备与设备之间无感发现和零等待传输创造了技术条件，设备注册到分布式软总线中，无须关心彼此的协议，协议互相兼容由分布式软总线解决。

分布式软总线可以解决很多目前体验不好的问题，例如：

- 智能家居场景：午睡的时候，可以将手机碰一下智能窗帘的外设开关，碰一下表示关闭窗帘，碰两下表示打开窗帘。这样就不需要在手机上打开控制窗帘的 App，然后找到控制窗帘的开关，这个过程可能需要 20 秒，使用碰一碰可以让整个流程非常自然而快速。智能窗帘硬件也不用自己设计操作系统，直接使用鸿蒙系统即可，这样可以极大地节省软件研发成本。
- 一回家，你的智能穿戴设备就可以接入家庭软总线中，各个设备之间不用任何设置，即可互相快速通信。例如，正在工作的扫地机器人，可以识别你的手机，发现主人回家了，机器人可以自己回到充电桩中。智能电灯发现主人回来了，可以自动打开灯光，这一切都不需要任何人为的操作。

2. 分布式设备虚拟化

分布式设备虚拟化平台可以将不同的设备连接起来，形成一个整体。例如，在一个手机上可以访问其他手机的摄像头，从而实现设备的资源融合，多种不同的设备共同形成一个超级虚拟终端。如图 1-3 所示。

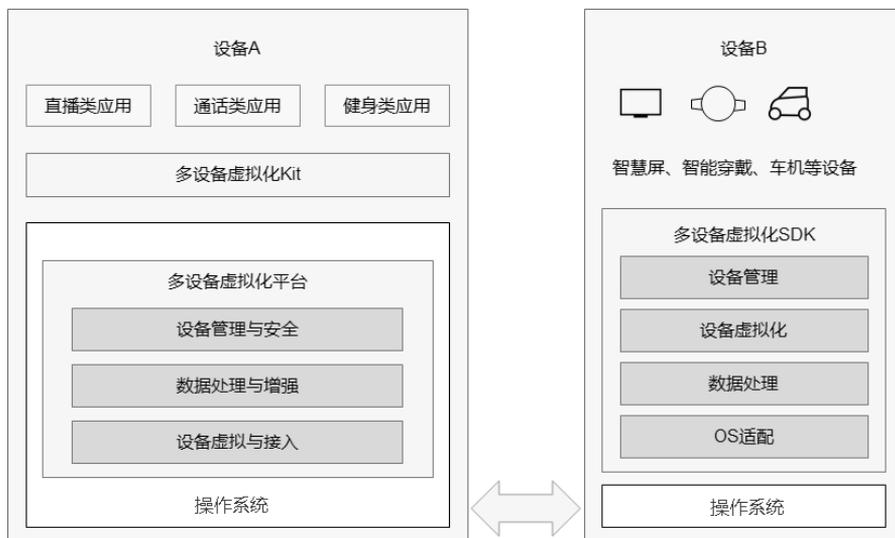


图 1-3 分布式设备虚拟化

这里以视频通话和游戏场景来说明分布式设备虚拟化的作用。

- 视频通话场景：在做家务时突然接收到视频电话，无法手持手机，使用手机不方便，这时候，可以将手机与华为智慧屏连接，并将智慧屏的屏幕、摄像头与音箱虚拟化为手机的本地资源，替代手机自身的屏幕、摄像头、听筒与扬声器，实现一边做家务、一边通过智慧屏和音箱来视频通话。智慧屏是华为的一种超级显示设备，后面再详细介绍。
- 游戏场景：在智慧屏上玩游戏时，可以将手机虚拟化为遥控器，借助手机的重力传感器、加速度传感器、触控能力，为玩家提供更便捷、更流畅的游戏体验。

3. 分布式数据管理

分布式数据管理基于分布式软总线的能力，实现应用程序数据和用户数据的分布式管理。所谓数据分布式管理是指，应用程序数据和用户数据不与某台设备绑定，数据可以在跨设备间进行访问。开发者不用关心数据存储在哪台设备，直接调用相应的模块就能访问到数据。如图 1-4 所示。

从图中可以看出，设备 A 中有设备 / 用户 / 应用 Profile、状态事件 / 位置、文本数据、媒体数据等数据，设备 B 中有设备 Profile、状态事件、文本数据。设备 A 和设备 B 都将数据同步到超级虚拟终端中，这样，超级虚拟终端就拥有设备 A 和设备 B 的所有数据，并且可以实时地同步更新。这就像设备 A 和设备 B 使用的同一块磁盘一样，任何一个设备对磁盘的更改，其他设备都能查看到。

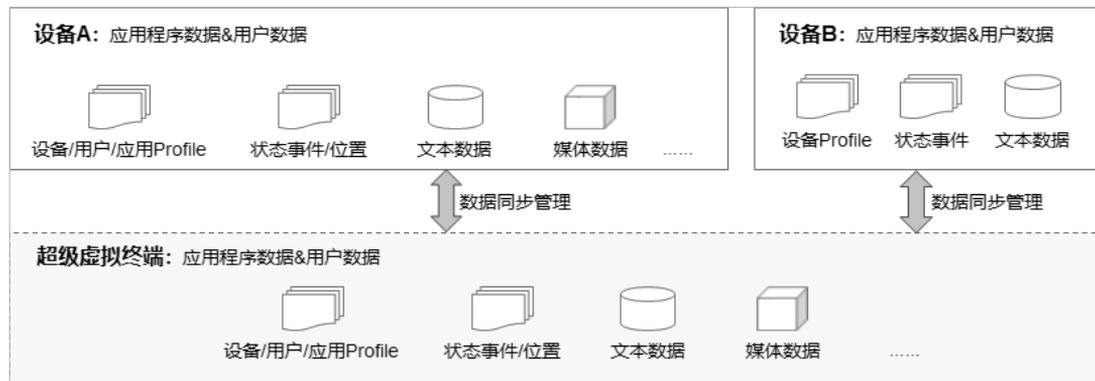


图 1-4 分布式数据访问

这里有一个概念——超级虚拟终端（Super Virtual Device），亦称超级终端，通过分布式网络技术将多个终端 / 设备的能力进行整合。这些终端的数据存放在一个虚拟的硬件资源池里，根据业务需要统一管理和调度终端能力，来实现对外提供服务，通过超级虚拟终端设备之间形成相互协同的工作场景。

例如，在搭载鸿蒙系统的手机上，可以直接向附近的计算机、平板电脑、手表、窗帘等，推送相关的数据，从而控制其他设备实现相应功能，这也是华为为鸿蒙系统“多端协同，多端流转”提供的一个重要能力。

超级虚拟终端将会带来十分强大的生态体验，能够将几乎所有搭载鸿蒙系统的电子设备融为一体，打造完全无缝的使用体验，让设备与设备之间的互联更为高效。

那么 Android、iOS 的系统不能互联，能够实现同样的功能吗？从某种程度上说，部分是可以的，只是无法达到鸿蒙的润物细无声的体验，同时硬件成本非常昂贵。

以 Android 为例，要将 Android 系统安装到窗帘上，窗帘就必须搭载一套成本相对较高的硬件，才能安装 Android 系统，从而实现窗帘与手机之间无缝的集成互联，如果窗帘的硬件较弱，无法安装 Android 系统，那么窗帘和 Android 手机就必须通过一些成本较低的蓝牙进行通信，并且 Android 手机上需要安装能接收和控制窗帘的 App，这样从体验上来说，连接过程比较烦琐、响应速度也会变慢一些，无法做到无感操作。

4. 分布式任务调度

分布式任务调度表示可以跨设备进行任务调度，任务可以通过远程调用，执行在其他设备上，从而实现一些只有其他设备才具有的能力。

分布式任务调度有如下使用场景。

- 长任务场景：在手机上正在处理一个视频文件，需要 20 分钟，可以把这个处理任务的信息发送到智能手表上，从而可以在智能手表上随时查看视频处理信息。

- 导航场景：如果用户驾车出行，上车前，在手机上规划好导航路线；上车后，导航自动迁移到车机和车载音箱；下车后，导航自动迁移回手机。如果用户骑车出行，在手机上规划好导航路线，骑行时智能手表可以接续导航。

1.4 鸿蒙系统的分层架构

在软硬件开发中，为了让一个复杂的系统能够顺利实现，常用的方法是分层设计，大家熟知的例子就是网络七层协议。鸿蒙系统也是分层设计的，从下到上有四层组成，分别是内核层、系统服务层、框架层和应用层。鸿蒙系统的技术架构如图 1-5 所示。



图 1-5 鸿蒙系统的技术架构

下面对这四层进行详细介绍。

1.4.1 内核层

内核层是操作系统的核心，主要是管理硬件资源，同时为外层应用提供系统调用。为了管理各种硬件，如显卡、声卡，需要有驱动子系统。为了管理进程、线程、内存等资源，需要有内核子系统。

内核子系统：鸿蒙系统采用多内核设计，支持针对不同硬件资源受限设备选用适合的操作系统内核。目前，鸿蒙系统包含 Linux 内核、LiteOS 内核。Linux 内核主要兼容 Android，由于 Android 是依赖于 Linux 内核的，所以为了支持 Android，必须要内置 Linux 内核。华为对 Linux 内核进行了优化裁剪，去掉了很多不需要的组件，同时添加了鸿蒙的一些新特性，如分

布式框架、分布式数据存储等。

LiteOS 内核主要支持小型物联网设备，可以根据硬件资源进行裁剪，LiteOS 内核还具有低功耗的特点，非常适合价格便宜的微型硬件设备。

驱动子系统包含了常用的驱动程序，其主要模块包括硬件驱动框架（**Hardware Driver Foundation, HDF**）。硬件驱动框架用于统一对外设进行管理，访问、修改、控制外设的状态。同时，硬件驱动框架还提供驱动开发的框架，驱动开发是开发工作中难度较大的部分，需要对操作系统及硬件接口及原理有较多的了解，才能写出一个完整的、正确的驱动。驱动开发的框架可以简化驱动的开发，让程序员不用学习太复杂的驱动原理，只需要在框架留出的代码桩处写代码，即可完成一个简单的驱动程序。

1.4.2 系统服务层

系统服务层主要提供系统的基础服务，为上层框架提供底层系统调用支持，包含了大量的基础服务组件。如分布式软总线、分布式数据管理、分布式任务调度等。其中方舟多语言运行时环境，能够支持多种编程语言的编译及运行，为系统支持多语言提供了帮助。

1.4.3 框架层

框架层主要为应用开发者服务，是系统对应用开放的接口。为了丰富开发生态，鸿蒙系统提供了多种语言的接口，目前支持 **Java/C/C++/JS** 等语言。同时框架层还提供了用户界面（**UI**）框架、用户程序框架、**Ability** 框架以及操作硬件服务的开发组件，开发者可以通过这些组件快速实现特定功能，从而加速应用的开发。

1.4.4 应用层

应用层包括系统应用和非系统应用。系统应用包括鸿蒙桌面、控制栏、系统设置、电话和短信等。非系统应用指第三方开发者开发的，可以通过应用商店下载安装的应用，如 **QQ**、微信、抖音、板栗看板等。本书主要围绕框架层、应用层，对软件开发进行讲解。

1.5 小结

现在，你应该感觉很平静，甚至充满了成就感，因为你开始逐步了解鸿蒙系统是什么了，以及鸿蒙系统能给世界带来什么。就像我们学写程序的时候，第一个程序永远是 **Hello**

World，鸿蒙以它先进的分布式系统设计理念，让万物互联更为容易，在不久的将来，软硬件生态会逐步跟上，那时候，我们日常生活中能体验到以前只能在科幻电影中出现的场景。每天下班回家，再也不用在指纹锁中录入指纹，只需要带上智能手表，轻轻地敲两下门，智能手表就可以自动和电子锁进行沟通，确认是谁回家了。通过超级虚拟终端技术，智能手表将电子锁当作了自己的外设，从而自动操作了电子锁。这种超级虚拟终端技术，是一种对用户透明，能让万物无感互联的令人兴奋的技术突破。

回到家后，智能家居能感应到你回到了家，空调会自动打开，窗帘会自动打开，冰箱的显示屏上也会提醒你晚餐可以吃什么，并根据你的喜好，在食物不够的时候，提前在电商平台下单。热水壶能将水调整到你喜欢的温度。

这些都是鸿蒙系统给我们的生活带来的各种可能，这些可能在很多科幻电影及我们的脑海里都出现过，只是互联万物的鸿蒙系统让这种可能离我们更近——未来可期！

第 2 章

搭载鸿蒙应用开发环境

在进行鸿蒙 App 开发之前，我们需要搭建一个鸿蒙 App 开发环境。本章将详细介绍鸿蒙 App 开发环境搭建相关知识，为下一步学习夯实基础。

2.1 开发环境简介

开发鸿蒙 App，华为提供了 HUAWEI DevEco Studio 集成开发环境，这个 IDE 面向全场景多设备，提供一站式开发平台，支持分布式原子化服务和应用的开发。在搭建 HUAWEI DevEco Studio 之前，需要确定鸿蒙 App 对开发环境的要求，具体要求如下，目前的计算机配置应该都能支持这个系统要求。在操作系统、SDK、IDE 三方面的具体要求如表 2-1 所示。

表 2-1 具体要求说明

项 目	版 本 要 求	说 明
操作系统	操作系统：Windows10 64 位或 macOS 10.14/10.15/11.2.2；内存：8GB 及以上；硬盘：100GB 及以上；分辨率：1280×800 像素及以上	内存越大运行越快
鸿蒙 App 开发 SDK	SDK API Version 5	当你阅读本书的时候，可能 SDK 已经升级到最新版本了，不过没关系，你仍然可以使用和本书同样版本的 SDK 来学习，学会之后，再升级到新版本的 SDK 会非常容易，一般来说 SDK 是向下兼容的
IDE	DevEco Studio 2.1 Release	当你看到本书的时候，可能有更新的版本，为了做练习时，能和本书保持一致，请使用这个版本，后面会列出该版本的下载地址

HUAWEI DevEco Studio 是基于 IntelliJ IDEA Community 开源版打造的集成开发环境。DevEco Studio 面向不同终端，可以开发多种面向不同设备的华为应用，可以开发智能手表、手机、智慧屏、计算机应用等。不同设备的应用界面效果如图 2-1 所示。



图 2-1 同一应用不同设备界面效果预览

DevEco Studio 为开发者提供了整套开发环境，开发者可以创建项目模板、开发、编译、调试及发布 HarmonyOS 应用。

和大多数开发环境一样，鸿蒙系统除了提供基本的开发调试能力外，针对鸿蒙系统自身的特点，DevEco Studio 也支持以下一些特性，其主要特点如图 2-2 所示。



图 2-2 DevEco Studio 的主要特点

下面简要地介绍一下这六个特点：

- 多设备统一开发环境。多种设备的应用开发都可以在 DevEco Studio 中完成。包括手机（Phone）、平板电脑（Tablet）、智慧屏（TV）、智能穿戴（Wearable），车机系统（Car）、轻量级智能穿戴（LiteWearable）和智慧视觉（Smart Vision）设备。
- 支持多语言的代码开发和调试。DevEco Studio 支持 Java、XML（Extensible Markup Language）、C/C++、JS（JavaScript）、CSS（Cascading Style Sheets）和 HML（HarmonyOS Markup Language）等语言，特别是支持前端开发，这样，很多前端程序员就可以开发移动端应用程序了。
- 支持 FA（Feature Ability）和 PA（Particle Ability）快速开发。通过工程向导快速创建 FA/PA 工程模板，一键式打包成 HAP（HarmonyOS Ability Package）。Feature Ability 是表示有页面的程序，用于提供与用户交互的能力，其实就是页面功能的一些类。Particle Ability 是粒子能力，表示一些微小的粒子功能，没有界面的程序，包含 Service Ability 和 Data Ability，分别用于编写服务和数据访问代码。
- 支持分布式多端应用开发。一个项目及一份代码可以在不同的设备上运行，支持在不同设备上显示实时界面，对应用进行多设备调试。
- 支持多设备模拟器。提供多设备的在线模拟器，包括手机、平板电脑、车机、智慧屏、智能穿戴设备的模拟器，方便开发者高效调试。
- 支持多设备预览器。提供 JS 和 Java 预览器功能，可以实时查看应用的布局效果，支持实时预览和动态预览；同时还支持多设备同时预览，查看同一个布局文件在不同设备上的呈现效果。

2.2 HUAWEI DevEco Studio 开发环境安装

HUAWEI DevEco Studio 支持 Windows 和 macOS 系统，两个系统中安装过程非常相似，安装 HUAWEI DevEco Studio 开发环境主要安装过程如图 2-3 所示。

搭建 HarmonyOS 应用开发环境主要包括软件安装和配置开发环境。

软件安装流程包括安装 DevEco Studio 开发环境，如果是 Windows 计算机那么安装 DevEco Studio 的 Windows 版本，如果是 Mac 计算机那么就需要安装 DevEco Studio 的 Mac 版本。

配置开发环境依赖一些软件包，需要配置一些包服务器，从服务器中下载一些必要的依赖包，其中设置 npm 代理、设置 Gradle 代理、设置 IDE 的代理是可选配置，可以在需要的时候再做配置。HarmonyOS SDK 是必须下载的依赖包，这也是开发鸿蒙应用必须要依赖的 SDK。

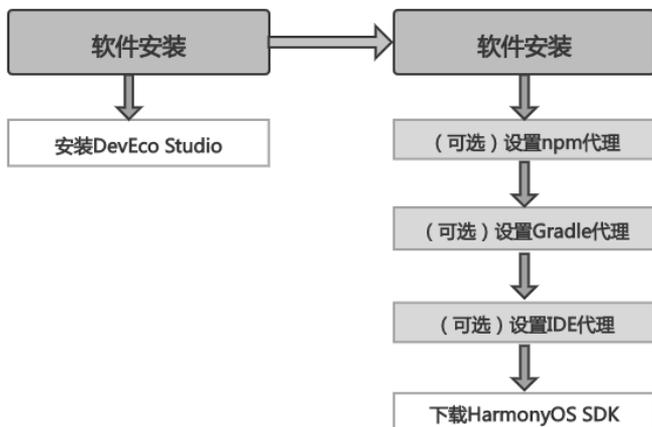


图 2-3 HUAWEI DevEco Studio 安装过程

2.2.1 macOS 系统中安装 HUAWEI DevEco Studio

以 macOS 系统安装 HUAWEI DevEco Studio 开发环境安装为例，主要步骤如下。

在鸿蒙官网 <https://developer.harmonyos.com/cn/develop/deveco-studio>（扫描右侧二维码）下载 macOS 系统的版本，该软件包大约 1040MB，如图 2-4 所示。



DevEco Studio 2.1 Release				
Platform	DevEco Studio Package	Size	SHA-256 checksum	Download
Windows(64-bit)	devecostudio-windows-tool-2.1.0.501.zip	890M	a22807ae2ef2d6f4afe5aa6d57b2714d66db0b2cb6674580e7aa6a26be68	↓
Mac	devecostudio-mac-tool-2.1.0.501.zip	1040M	70b99e1f8874ec9cd38c6c286589061f0f22a227b5f97e04f7823620890a9	↓

图 2-4 Windows 和 Mac 软件包具体信息

下载完成后，双击下载的“deveco-studio-xxxx.dmg”软件包，xxxx 是软件包的版本号，本书使用的是 2.1.0.501 这个版本，你也可以使用更新的版本。

在安装界面中，将“DevEco-Studio.App”拖动到“Applications”中，等待安装完成，如图 2-5 所示。

大概需要 1 分钟，DevEco Studio 就安装完成了。可以在应用程序中找到 DevEco-Studio 图标，双击即可启动开发环境。

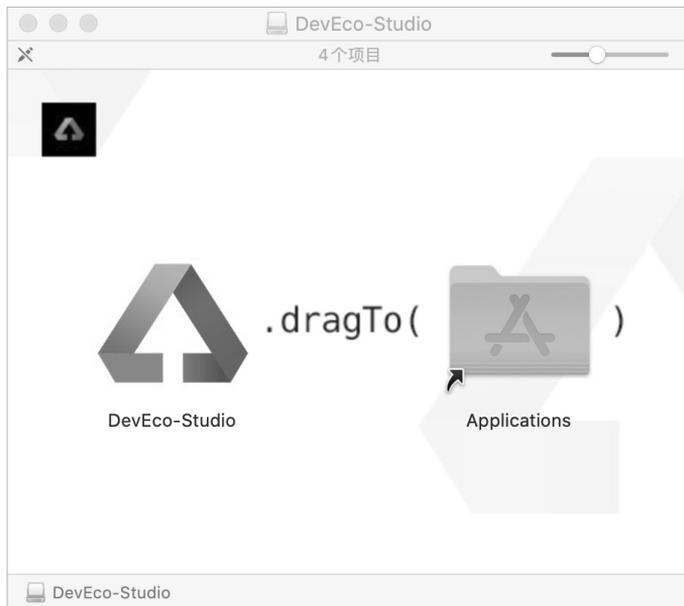


图 2-5 DevEco-Studio.App 安装图示

2.2.2 Windows 系统中安装 HUAWEI DevEco Studio

Windows 上 DevEco Studio 安装比 macOS 上安装步骤多一些，具体如下。

在鸿蒙官网 <https://developer.harmonyos.com/cn/develop/deveco-studio#download> 下载 Windows 系统的版本（扫描右侧二维码），该软件包大约为 890MB，如图 2-6 第二行所示。



鸿蒙官网

DevEco Studio 2.1 Release				
Platform	DevEco Studio Package	Size	SHA-256 checksum	Download
Windows(64-bit)	devecostudio-windows-tool-2.1.0.501.zip	890M	a22807ae2ef2d6f4afe5aa6d57b2714d66db0b2cb6674580e7aa6a26be	↓
Mac	devecostudio-mac-tool-2.1.0.501.zip	1040M	70b99e1f8874ec9cd38c6c286589061f0f22a227b5f97e04f7823620890:	↓

Release版本提供更稳定的功能特性，点击查看版本说明。

图 2-6 Windows 和 Mac 软件包具体信息

下载完成后，双击下载的“deveco-studio-xxxx.exe”软件包，xxxx 是软件包的版本号，本

书使用的是 2.1.0.501 这个版本，你也可以使用更新的版本。双击进入安装向导，提示 DevEco Studio 将要开始安装，建议你关闭其他程序。安装向导界面如图 2-7 所示。

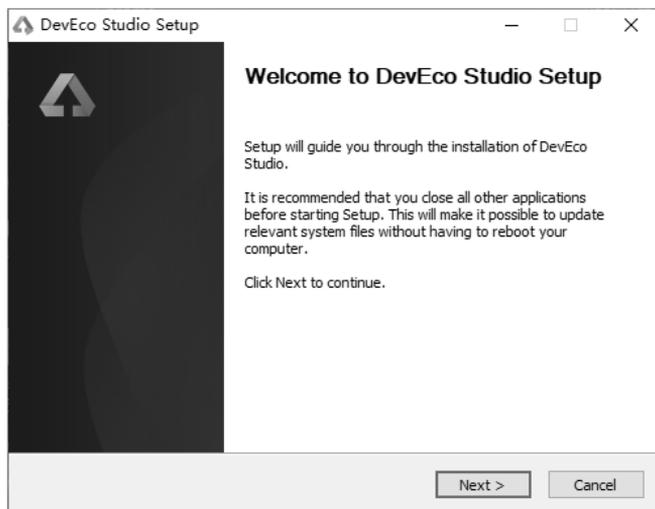


图 2-7 DevEco Studio 安装向导界面

设置安装路径，安装目录大约需要 1.7GB 的剩余空间，建议不要安装在系统盘中，系统盘的空间有限，不利于以后安装 DevEco Studio 的其他插件及 SDK。设置安装目录界面如图 2-8 所示。

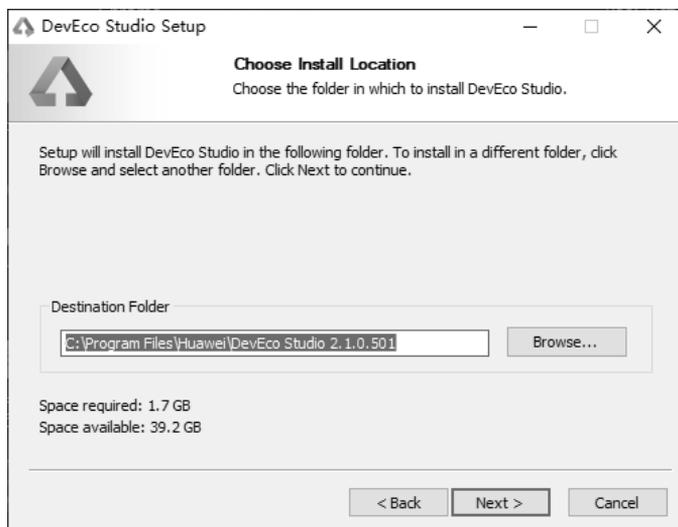


图 2-8 DevEco Studio 安装 - 设置安装目录界面

配置安装选项，这里有三个安装选项，具体如下。

- 创建一个桌面快捷方式；

- 添加一个环境变量到 PATH 中；
- 在右键上下文菜单中添加“Open Folder as Project”。

建议这几个选项都选择上，如图 2-9 所示。

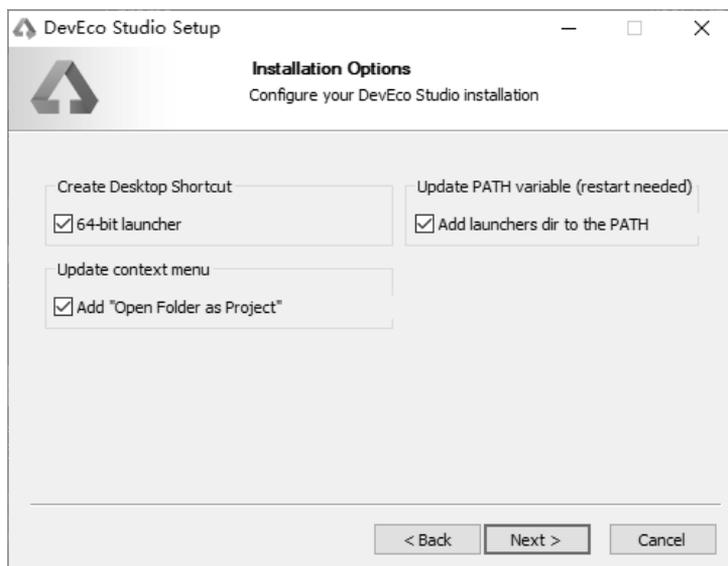


图 2-9 DevEco Studio 安装 - 配置选择界面

在开始菜单中添加启动快捷文件夹，如图 2-10 所示。

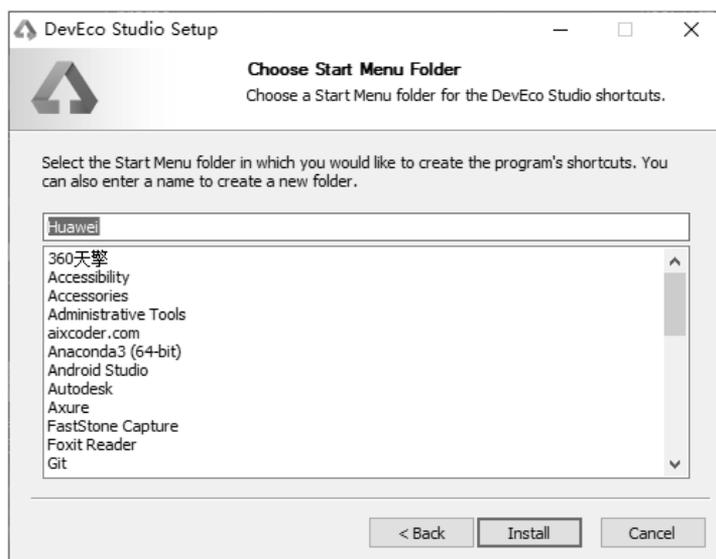


图 2-10 DevEco Studio 安装 - 添加快捷方式界面

单击 **Install** 按钮，进入安装界面，如图 2-11 所示。

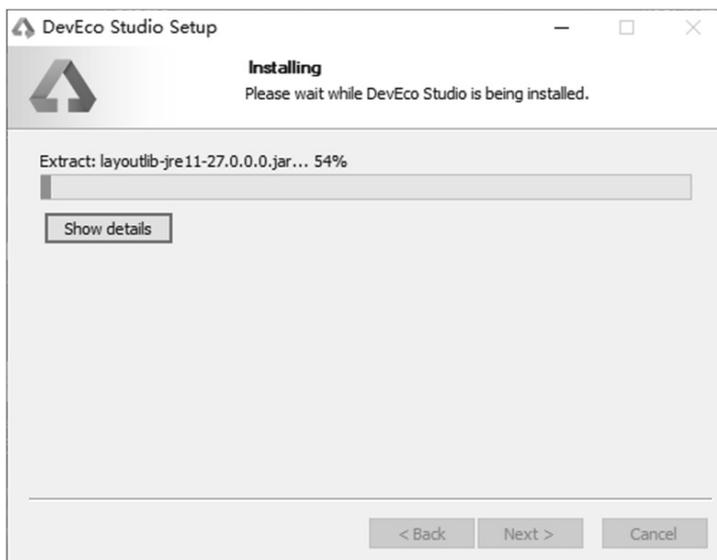


图 2-11 DevEco Studio 安装 - 运行安装界面

安装完成后，最好重启一下系统，如图 2-12 所示。

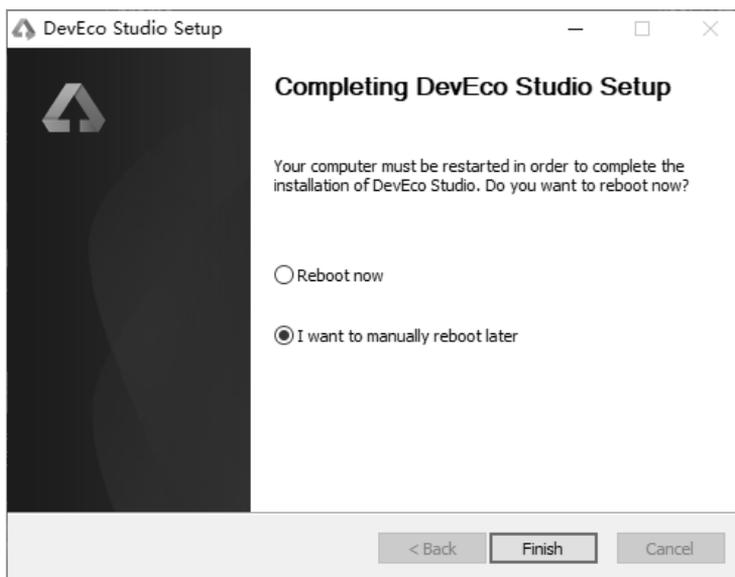


图 2-12 DevEco Studio 安装 - 安装完成界面

2.3 配置 DevEco Studio

安装完成 DevEco Studio 后，第一次启动该程序，会引导我们进行环境设置，主要设置步

骤如下，这里以 macOS 版本为例，与 Windows 版本类似。

提示是否创建启动脚本，启动脚本用于从命令行启动 DevEco Studio 开发环境。选择“Create a script opening files and projects from the command line”，表示可以从命令行打开一个项目或文件。为了方便，我们通常选择这个设置，如图 2-13 所示。

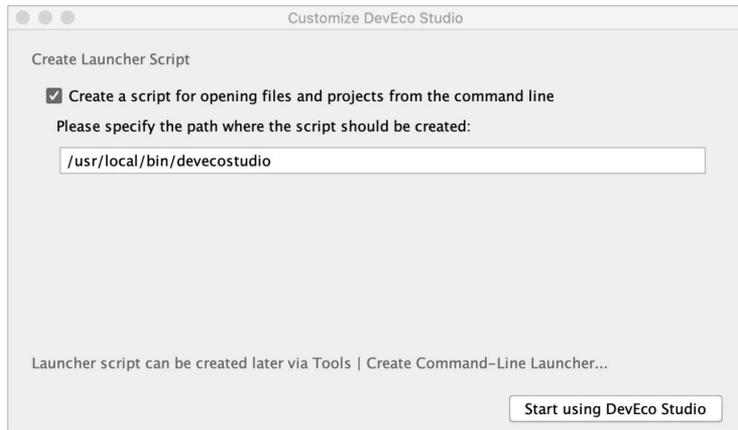


图 2-13 DevEco Studio 创建启动脚本并配置命令行启动

选择这个选项后，在命令行执行 `devcostudio` 时即可快速打开 DevEco Studio 开发环境。

下一步是展示 HUAWEI DevEco Studio 的用户协议，主要是介绍 DevEco 的一些功能，以及用户是否同意华为的一些用户隐私协议，这一步必须选择“同意”，否则程序将会退出，如图 2-14 所示。

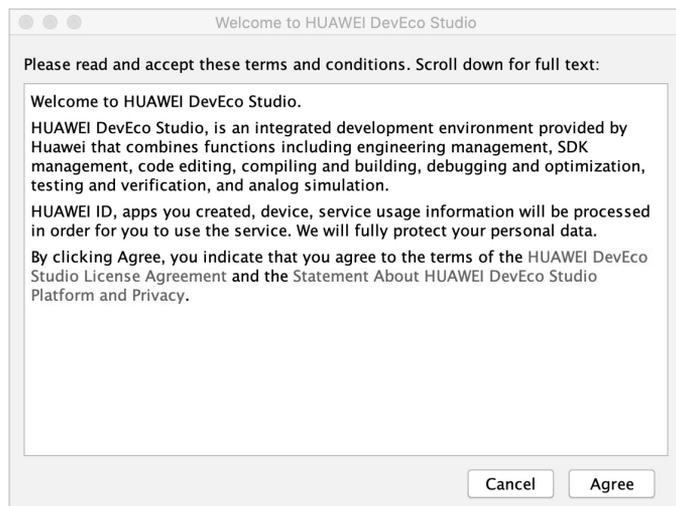


图 2-14 使用 DevEco Studio 需要同意的协议

有兴趣的读者可以看一下这个协议。

DevEco 运行需要 npm 工具及仓库，这里是配置 npm 仓库，默认为华为提供的仓库，如果你有其他仓库，可以更改，一般情况下不需要更改。npm 的配置信息存储在你的计算机的 /Users/ 当前用户 /.npmrc 文件中。配置 npm 仓库如图 2-15 所示。

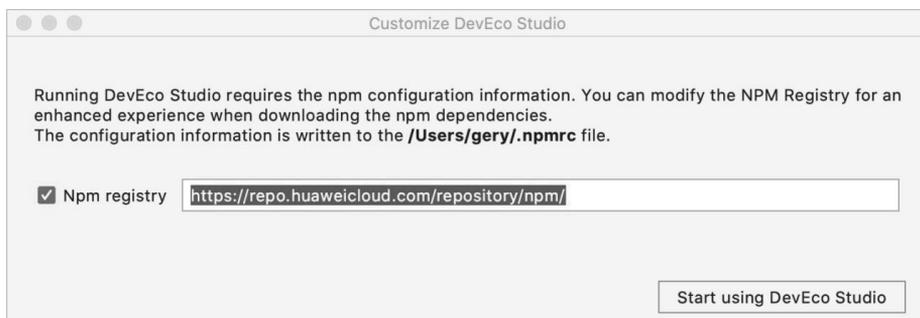


图 2-15 DevEco Studio npm 仓库配置

安装 HarmonyOS SDK，开发 Harmony 应用必须要安装鸿蒙 SDK，默认安装在 /Users/ 当前用户 /Huawei/sdk 目录中，目前，SDK 需要 636MB 空间。这一步会从互联网上下载 SDK，需要一些时间。注意，这个目录也包含了很多工具命令，后续会使用到这些工具命令。配置 SDK 如图 2-16 所示。

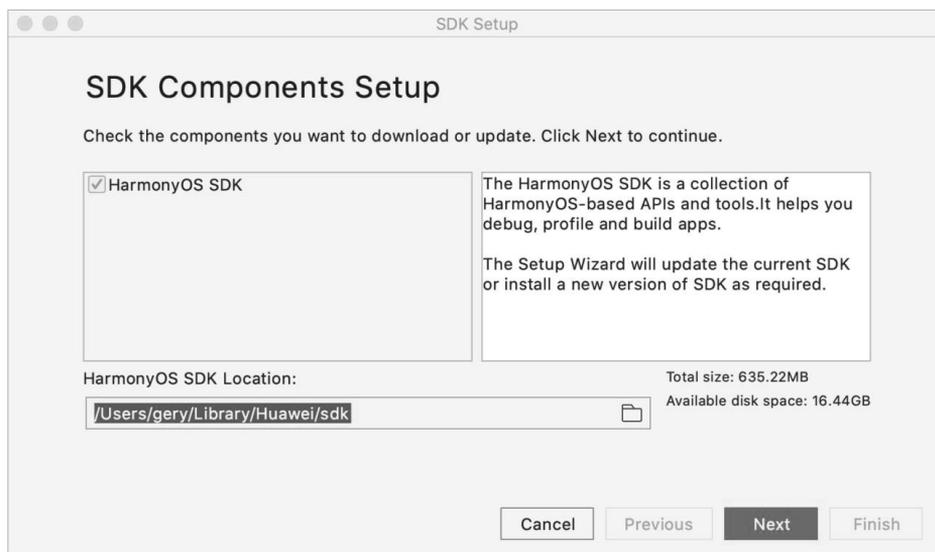


图 2-16 DevEco Studio 的 SDK 配置

这一步是配置信息确认，前面的是相关的配置，但是还没正式执行配置，这里需要确认一下先前的配置是否正确，如果不正确，可以单击 Previous 按钮进行修改。确认配置界面如图 2-17 所示。

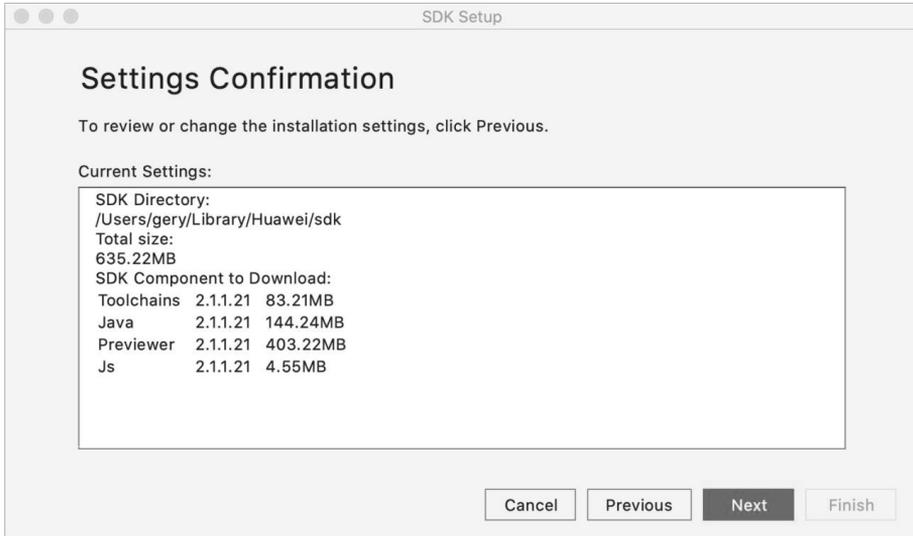


图 2-17 DevEco Studio 配置的最后确认

是否同意授权，单击 **Accept** 和 **Next** 按钮继续安装，如图 2-18 所示。

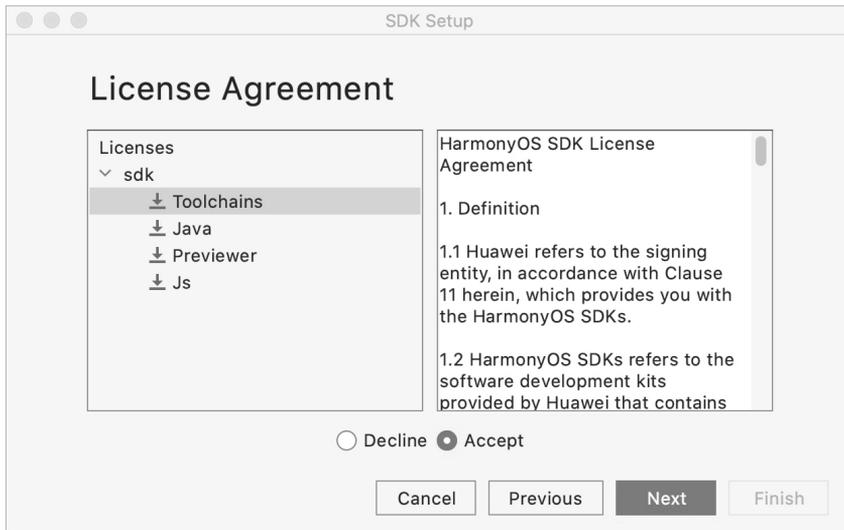


图 2-18 DevEco Studio 接受下载配置

这一步开始下载一些开发环境需要的工具链、SDK 等。下载完成后单击 **Finish** 按钮，整个配置过程就完成了。完成配置后下载界面如图 2-19 所示。

配置完成后，显示创建项目界面，这样就可以开始开发程序了。正式开发界面如图 2-20 所示。

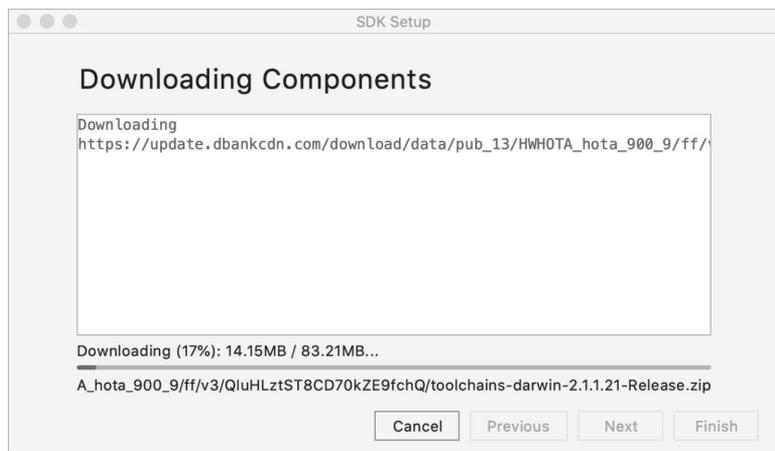


图 2-19 DevEco Studio 配置下载

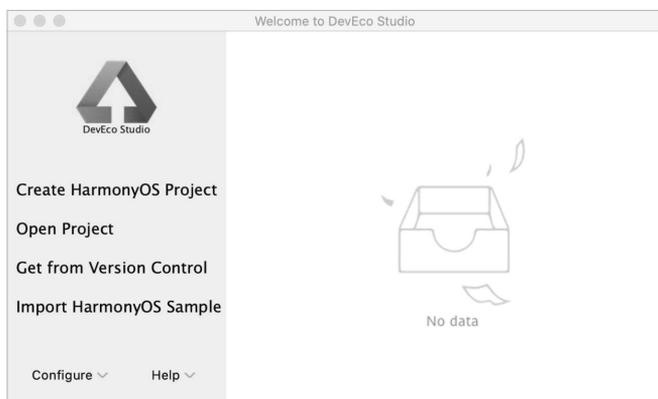


图 2-20 DevEco Studio 开始开发界面

2.4 小结

工欲善其事必先利其器，现在我们有了一个好的开发工具了。回想这些年的技术发展，日新月异。十几年前，开发 Android 程序的时候，只能用 Eclipse 的 Android 开发插件，直到 2013 年 5 月份，Google 在 I/O 开发者大会上才推出了基于 IntelliJ IDEA Java IDE 上的 Android Studio，从此，Android 才有了自己专用的开发工具。相比 Android 的开发人员，鸿蒙的开发人员是幸运的，DevEco Studio 及模拟器的运行速度非常快，相比最初的 Eclipse 和 Android Studio 来说，工作效率提升了数倍。遥想当年，Eclipse 和 Android Studio 的模拟器相当慢，甚至卡顿假死，很多开发者都是通过真机来调试，才能间接解决开发效率低的问题，而开发鸿蒙程序，你不会有这些担忧，这也是这些年国产系统的一大进步。

第 3 章

创建第一个鸿蒙应用

开发一个优秀的鸿蒙应用，需要大量的学习和编程实践。不过开始我们的第一个鸿蒙应用非常简单，只需要结合上一章安装好的开发环境，跟着本书一步一步地操作，就能很快实现。

3.1 第一个应用实现的目标

开始这一章的学习之前，我们必须安装 DevEco Studio 开发环境，具体安装方法请见上一章的讲解。为了简单，让大家快速入门，本章要实现的应用程序只有一个用户界面元素，如图 3-1 所示。

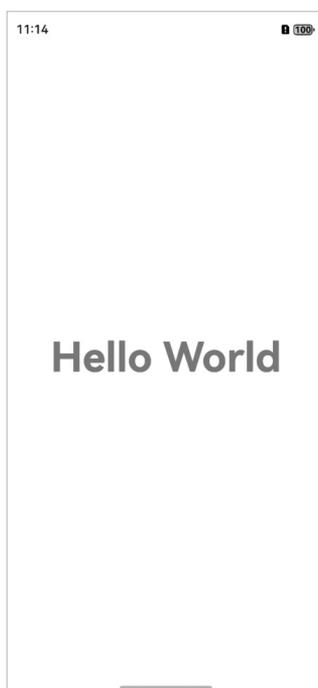


图 3-1 应用程序开始界面

本章代码可以在 `HelloWorld` 目录中找到。

这个程序有 1 个元素：文本标签（`Hello World`）。

第一个应用是不是非常简单？它仅仅是一个开始，下面我们一步一步来创建这个程序。新建一个项目的主要步骤如下。

首先打开 `DevEco-Studio`，该程序默认安装在“/应用程序”中。本书大部分案例是在 `macOS` 上开发，`Windows` 上的开发和在 `macOS` 上的开发基本一样。`DevEco Studio` 打开后，会进入 `Welcome to DevEco Studio` 界面，如图 3-2 所示。

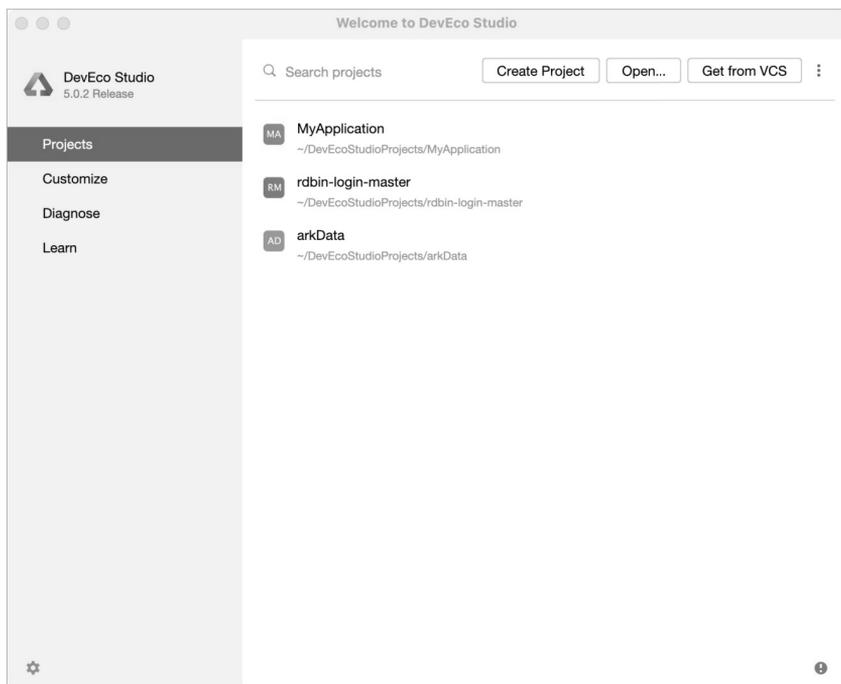


图 3-2 DevEco Studio 启动向导界面

顶部三个按钮和一个省略按钮的含义如下。

- **Create Project:** 创建一个新的鸿蒙项目，这个菜单是以向导式的方式创建项目。
- **Open Project:** 打开一个已经存在的项目，也就是我们从网上下载的项目，可以从这里打开。
- **Get from Version Control:** 从代码版本管理仓库中获取项目源文件。目前支持 `Git`、`Mercurial`、`Subversion` 等源码管理仓库。
- **Import Sample:** 导入鸿蒙的案例代码，鸿蒙提供了很多 `Demo` 案例代码，便于我们学习。

为了快速入门，这里创建一个新的项目，单击“`Create Project`”按钮打开“`Create`

Project” 弹出框，这个弹出框要求我们选择一个项目模板，如图 3-3 所示。

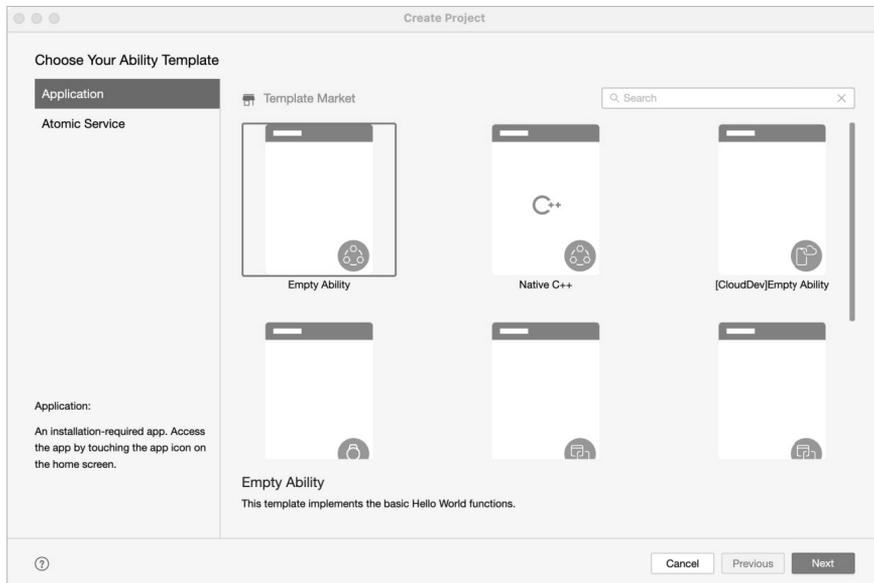


图 3-3 选择模板

在选择项目模板后，我们进入项目配置弹出框“Configure Your Project”，这个弹出框主要是用来设置项目的一些基本属性，如图 3-4 所示。

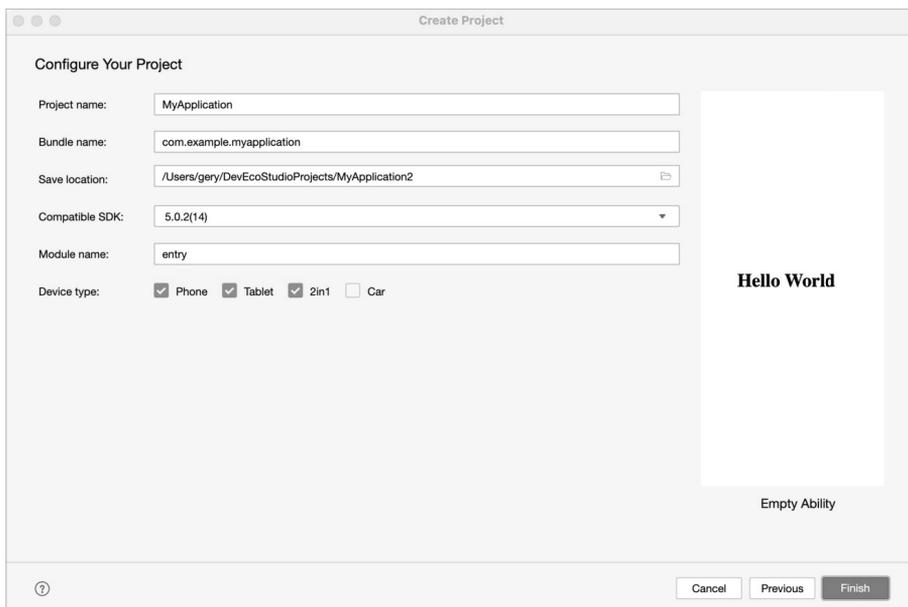


图 3-4 项目属性设置

需要设置的基本属性如下。

- **Project Name:** 是项目的名字，项目的名字不能有中文，只能是字母、数字、下画线。
- **BundleName:** 是包名，包名是为了区分不同的项目，由字母、数字、点、下画线组成，本书大多数案例都是以 `com.hellogdemos` 为包名，一般是一个域名反着书写的形式，反着写是为了避免重复。
- **Save Location:** 表示这个项目存放在本地计算机的位置。
- **Compatible SDK:** 是兼容的 API 版本，这里建议选择最新版本的 API。
- **Device Type:** 表示应用程序可以运行在哪些设备上，这里有 5 种设备，分别是手机（Phone）、平板电脑（Tablet）、电视（TV）、可穿戴电子产品（Wearable）、车载设备（Car）。如果你选择了多个设备类型，那么程序就要考虑在多个设备间兼容。

最后，单击 **Finish** 按钮，完成项目的创建。

项目创建完成之后，稍等一会儿，就会出现如图 3-5 所示的界面。左边是项目文件导航，右边是编辑区。

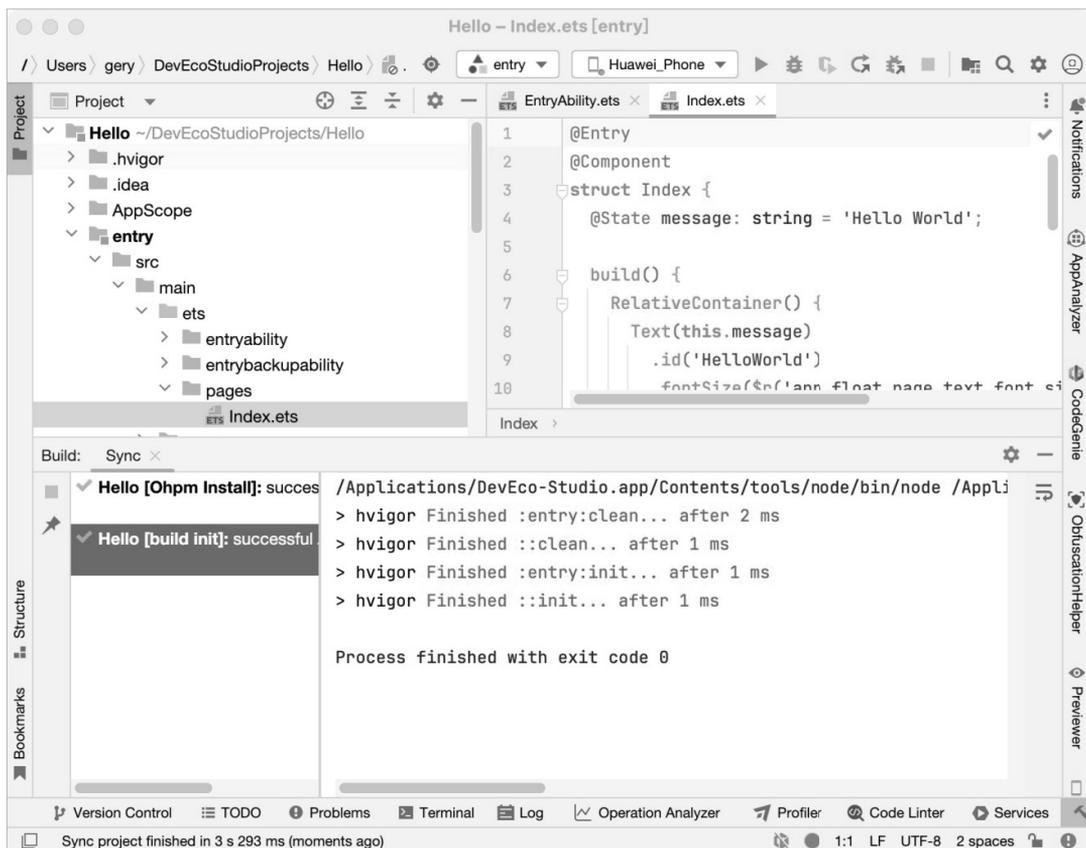


图 3-5 新项目目录结构

3.2 注册鸿蒙开发者账号并在模拟器上运行

新建完项目后，怎么运行这个程序呢？首先需要启动一个模拟器，然后让程序发布到模拟器上运行。启动模拟器需要注册一个华为开发者账号，下文讲解怎样注册华为开发者账号。

首先尝试启动模拟器，会提示我们登录为账号，单击菜单 **Tools>Device Manager**，出现界面如图 3-6 所示。

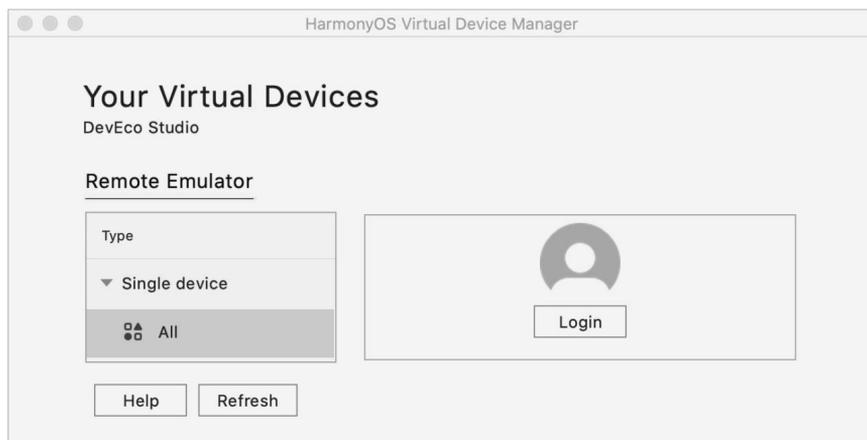


图 3-6 启动模拟器

单击 **Login** 按钮开始注册，单击注册链接。注册 / 登录界面如图 3-7 所示。



图 3-7 注册 / 登录华为账号

注册成功后，重新单击菜单 **Tools>Device Manager** 打开模拟器，会提示你“HUAWEI

DevEco Studio 想要访问您的华为账号”，从而获取华为账号中的一些信息，这里单击“允许”，如图 3-8 所示。



图 3-8 提示访问华为账号界面

单击“允许”后，会提示“您已成功登录客户端 HUAWEI DevEco Studio。”登录成功界面如图 3-9 所示。



图 3-9 成功登录华为账号

注册成功后，还需要实名认证，认证步骤比较多，这也是为了安全起见。通过实名认证，可以有效地避免开发一些病毒程序，让其他人使用。回到 HarmonyOS Virtual Device Manager

界面，提示 **The remote emulator requires real-name authentication**，如图 3-10 所示，意为远程模拟器需要一个用户真实身份认证。单击 **Go Authentication** 开始真实身份认证。

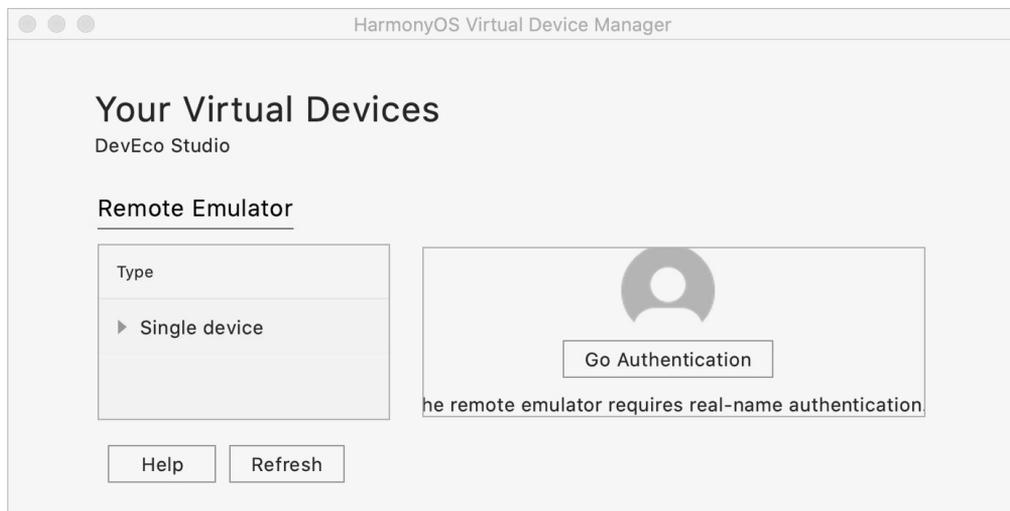


图 3-10 华为账号实名认证

然后弹出如图 3-11 所示的界面，英文表示：首先请单击 **here** 链接开始实名认证，如果已经完成了实名认证，请关闭 **DevEco Studio**，重新登录。这里我们没有进行实名认证，所以单击 **here** 链接。

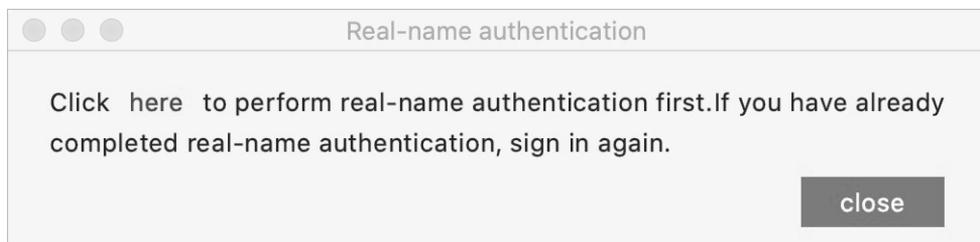


图 3-11 进入实名认证界面

浏览器自动打开“开发者实名认证”窗口，需要实名认证，才能成为华为开发者联盟合作伙伴，获得更多开发、分发应用等服务权益。这里我们选择个人开发者，如果你是企业，可以选择企业开发者，企业开发者需要上传营业执照等信息。选择认证方式界面如图 3-12 所示。

这里进入“开发者实名认证”界面的第二步，按照要求选择“是”“否”即可。这里选择“否”，表示我们的应用不涉及游戏、金融、新闻社交等监管领域的应用，这样更容易认证通过。具体选择界面如图 3-13 所示。



图 3-12 认证方式选择界面



图 3-13 认证选择应用领域

进入银行卡或者身份证人工审核认证，个人银行卡认证只需要 3 分钟，身份证需要人工审核，审核时间为 1 ~ 2 个工作日，这里我们选择了银行卡认证，因为时间就是效率，如果你介意用银行卡认证，那就准备一下身份证。单击“前往认证”即可进入下一步。两种认证界面如图 3-14 所示。

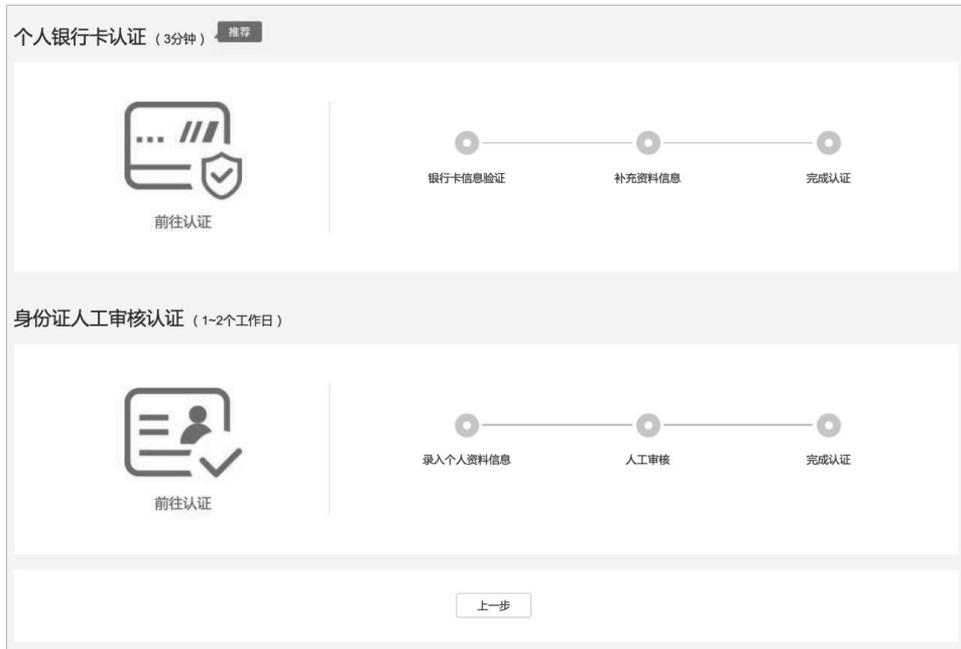


图 3-14 认证方式选择界面

这一步需要填写你的真实姓名、身份证号码、银行卡号及在该银行预留的手机号码，然后填入手机验证码，单击“下一步”即可，如果填写错误，会提示你认证错误。认证信息填写界面如图 3-15 所示。

银行卡信息验证 补充资料信息 完成认证

个人银行卡实名认证 (3分钟)

1. 如何进行个人银行卡认证?
2. 个人银行卡认证支持个人借记卡及信用卡。若您使用交通银行、北京银行、上海银行、广发银行、浦发银行或者平安银行卡进行个人实名认证,需要先开通此银行的银联在线支付。
3. 个人银行卡信息仅用于实名认证,您的银行卡信息将受到严格保护。

*真实姓名: 请输入身份证上的姓名

*身份证号码: 请输入您的身份证号

*银行卡号: 请输入您的银行卡号

*联系人手机: +86 请输入该卡在银行预留的手机号码

上一步 下一步

图 3-15 认证信息填写

上一步通过后，会要求我们填写一些个人详细信息，如邮箱、家庭住址等，填写完成后，单击“下一步”，如图 3-16 所示。

开发者实名认证

完善更多资料

* 真实姓名： 已验证

* 身份证号码： 已验证

* 银行卡号： 已验证

* 联系人手机：

* 联系人邮箱：

* 邮箱验证码： 94 秒后重新获取
验证码已经发送到您的邮箱。

* 所在地区： 省份 城市

* 地址：

* 获知渠道： 微信 微博 论坛 沙龙 开发者大赛
 其他

图 3-16 补充资料完善

如果一切顺利，到这一步就代表我们实名认证成功了。认证成功界面如图 3-17 所示。

银行卡信息验证 完成 补充资料信息 完成 完成认证 完成

实名认证成功

欢迎您加入华为开发者联盟，您现在可以使用华为提供的丰富服务。

用户名：176****2842
类型：个人用户
[点击了解更多联盟服务>>](#)

[管理中心](#) [返回首页](#)

图 3-17 认证成功

实名认证成功后，重启 DevEco Studio，然后单击 Tools 菜单中的 Device Manager。如图 3-18 所示。

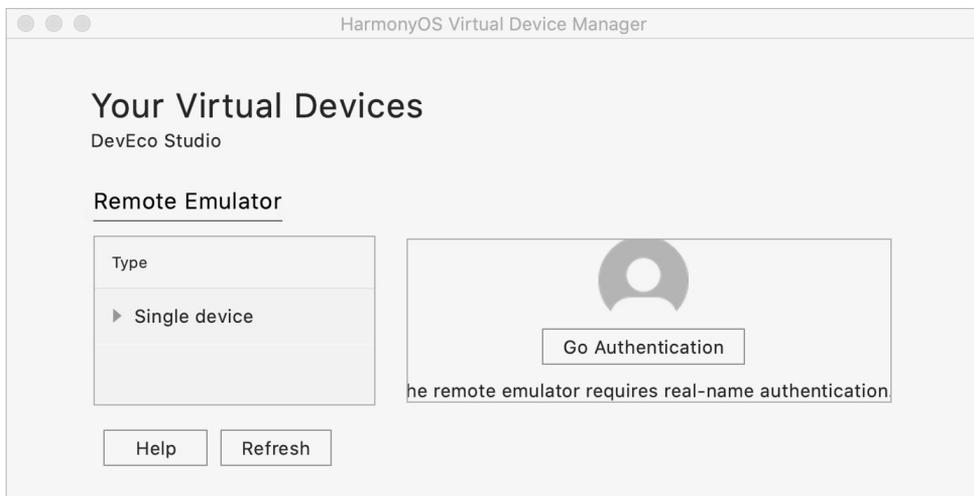


图 3-18 进入授权

单击 Go Authentication 进入授权页面，如图 3-19 所示。



图 3-19 授权页面

单击“允许”，进入 HarmonyOS Virtual Device Manager 界面，界面中已经显示了设备（如 P40），如图 3-20 所示。

选择一个你感兴趣的模拟器，单击绿色的运行按钮，即可启动模拟器。华为手机虚拟效果如图 3-21 所示。

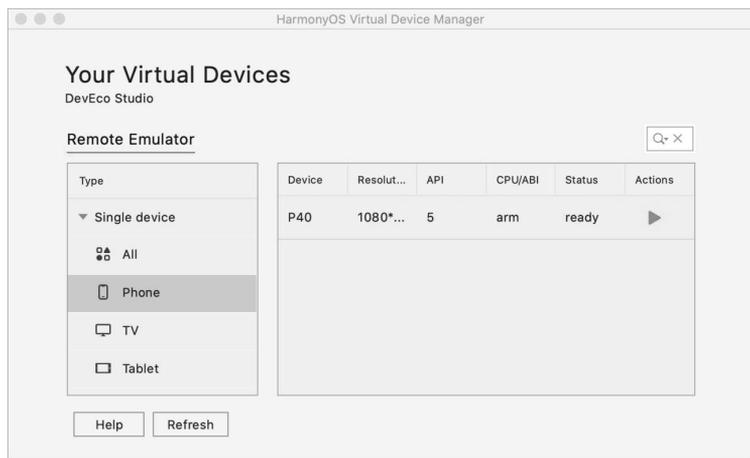


图 3-20 虚拟设备管理



图 3-21 华为模拟器设备

3.3 使用真机运行程序

在研发中，除了使用模拟器调试程序外，也需要使用真机对程序进行调试，模拟器在性能、体验、相机、话筒的测试上比不上真机，这时候使用真机能最大限度地提高生产力。

为了安全性，使用真机的调试比较复杂，不是说把手机用 USB 数据线连接到 DevEco Studio 开发环境中，DevEco Studio 就能够调用手机进行断点调试了，必须进行一个相对复杂的配置、证书生成等过程，才能够在真机上进行调试。为什么华为限制这么严格呢？原因是华为想每一个在真机上运行的程序，都需要受到华为安全体系的控制，这样开发病毒或不安全的软件就没那么容易了，从而能够最大限度地保证安全性。

真机调试有如下五个步骤。

- 使用 DevEco Studio 生成证书请求文件。
- 申请应用调试证书和设备注册。
- 申请项目和应用。

- 在开发环境中配置相关信息。
- 运行程序。

3.3.1 使用 DevEco Studio 生成证书请求文件

首先，需要在 DevEco Studio 中生成证书，证书是一个后缀为 p12 的文件，每一个证书都可以生成一个 .p12 文件，这个文件是一个加密文件。只要知道其密码，就可以供给所有的鸿蒙设备使用。这个证书一般用于开发调试，这个证书的目的，主要是限定开发主机的安全性。

打开 DevEco Studio 开发环境，依次选择菜单 Build>Generate Key and CSR，打开生成 .p12 文件的界面，如图 3-22 所示。

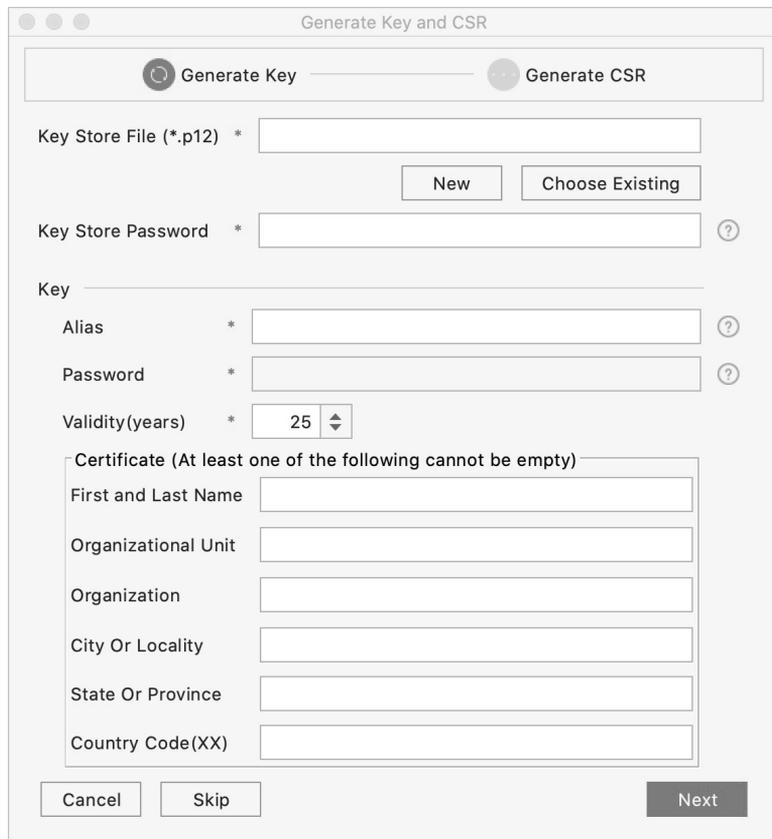


图 3-22 p12 证书生成界面

然后单击 New 按钮，通过密码生成一个 .p12 文件，.p12 文件只需要一个加密密码，如图 3-23 所示。

该界面中有三个输入框需要填写。

- **Key Store Path:** 选择密钥库文件的存储路径和文件名,只能是英文路径,不能包括中文,后缀是 p12。

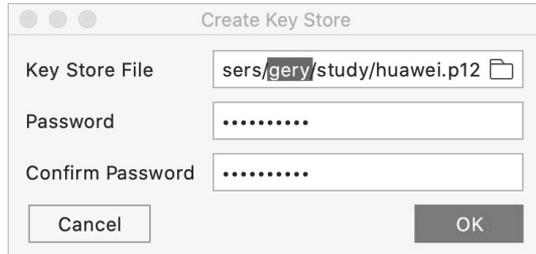


图 3-23 创建 .p12 文件

- **Password:** 设置密钥库密码,我们称为 p12 的,需要包含大小写字母,这个密码要记住,后续会经常使用。
- **Confirm Password:** 再次输入密钥库密码。

定义了 .p12 文件需要的信息后,其他内容根据实际情况填写即可,如图 3-24 所示。



图 3-24 证书信息完善

这里的信息虽然太多,但是也不复杂,只要按照要求填写即可。

- **Alias:** 别名,表示这个证书的别名,这个别名主要是为了更方便记忆。

- **Paasword:** 和上面的密码一致。
- **Validity:** 密钥文件的有效期，随便写。
- 其他是一些认证信息，按照要求写就可以了。

然后单击 **Next**，设置 CSR 文件的路径及文件名，最好和刚才的 .p12 文件放在一个目录，方便查找，如图 3-25 所示。

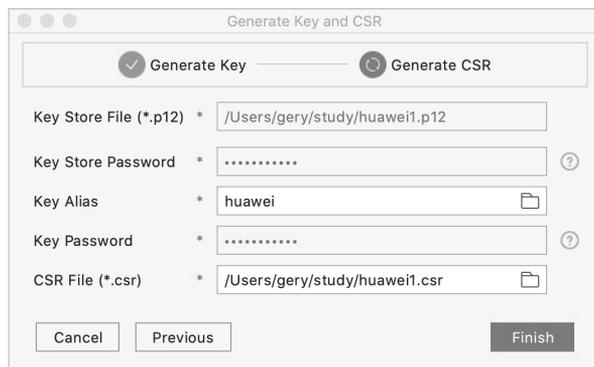


图 3-25 CSR 文件生成

单击 **Finish** 按钮，即可生成 .p12 和 CSR 文件了，这些文件在后面会使用到。

3.3.2 申请应用调试证书和设备注册

在本地生成了密钥文件后，同时需要在华为官网上也申请一些证书。打开 AppGallery Connect 网站，在华为官网上申请真机调试的证书，地址是 <https://developer.huawei.com/consumer/cn/service/josp/agc/index.html>（扫右侧二维码），注册一个账号并登录进去，注意，这里的账号和你的 DevEco Studio 开发环境的登录的账号应该是一样的。单击“证书、APP ID 和 Profile”，如图 3-26 所示。



图 3-26 进入华为官网申请证书

在左侧导航栏选择“证书管理”，进入证书管理页面，单击“新增证书”。如图 3-27 所示。

填写证书的名称，证书类型选择调试证书，然后将刚才在本地生成的 CSR 证书上传上去。如图 3-28 所示。



图 3-27 新增证书

图 3-28 使用 csr 申请测试证书

单击“提交”按钮，即可生成证书，证书生成后，单击“下载”按钮，即可将证书保存下来，供后续使用，这个证书的后缀名是 cer。每个人只能申请 2 个证书，证书的有效期限是 1 年，1 年后，需要重新申请。证书信息如图 3-29 所示。



图 3-29 证书有效时长

然后就是注册设备，在左侧导航栏里面选择“设备管理”，单击“添加设备”按钮，如图 3-30 所示。



图 3-30 添加开发设备

然后添加你自己的真机作为设备，如图 3-31 所示。这里的名称随便填写，类型根据你的真机类型进行选择，我使用的是手机，所以选择手机。设备管理的目的，是让华为知道我们的真机需要用于开发环境，以做一些安全权限上的解锁。



图 3-31 添加开发设备设置其信息

图中的名称、类型都比较好填写，UDID 是设备唯一标志，需要通过命令行工具才能获得。找到你计算机上的鸿蒙 SDK 的目录，我的目录在 `/Users/musk/Library/Huawei/sdk/toolchains` 中，这个目录中有一个 `hdc` 程序。将你的手机与计算机连接，然后运行：

```
#!/hdc shell bm get --udid
error: no devices/emulators found
```

如果显示 `error: no devices/emulators found` 错误，那么将会无法获取 UDID，请检查你的手机是否插入计算机，如果插入了，那么检查手机的 USB 调试模式是否打开，必须打开 USB

调试模式才能获取相应的 UDID。把下面的一些关于 USB 调试的选项都打开，就可以访问到 UDID 了，如图 3-32 所示。



图 3-32 打开 USB 调试模式

然后再次执行 `hdc` 命令：

```
# ./hdc shell bm get --udid
80E0EFE318C5231F1D84F47C5231F1D8996E8C5231F1D84F47289989BD77
```

这次执行成功了，将 `80E0EFE318C5231F1D84F47C5231F1D8996E8C5231F1D84F47289989BD77` 填入上面的输入框中，就可以注册设备了。设备注册成功界面如图 3-33 所示。



图 3-33 设备注册成功界面

3.3.3 申请项目和应用

申请了证书后，需要在华为网站上注册一个项目和应用，一个项目中可以包含多个应用。

这个应用就是我们实际开发的应用，这样华为官网就知道我们有一个应用需要在某台真机上运行了。

首先是创建一个项目，然后在项目中创建一个调试应用，单击“添加项目”按钮。如图 3-34 所示。

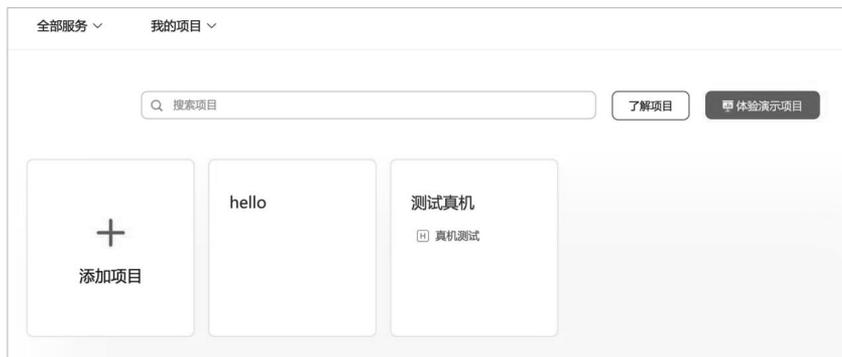


图 3-34 创建项目

输入项目名，可以为中文，也可以为英文，代表项目的名字，如图 3-35 所示。



图 3-35 填写创建项目名称

项目创建成功后，单击左边的菜单 HAP Provision Profile，如图 3-36 所示。

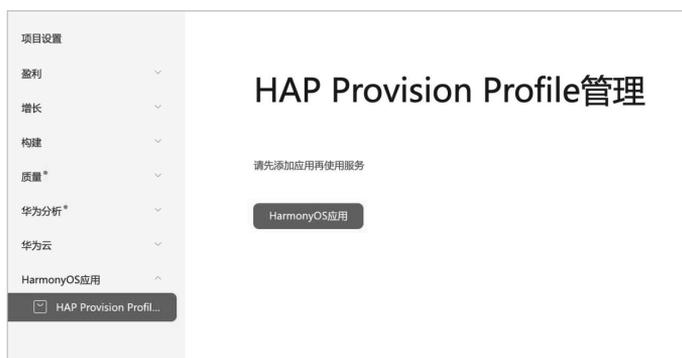


图 3-36 在项目中创建应用

然后创建应用，如图 3-37 所示。

图 3-37 填写创建应用信息

填写下面的内容。

- 选择平台：这里选择 App(HarmonyOS)。
- 支持设备选择手机，因为我的真机是手机。
- 应用名称随便填写。
- 应用包名，就是我们的应用的包名。
- 应用分类选择应用，因为我们不是游戏。
- 默认语言选择简体中文。
- 应用包名是需要调试的程序的包名，包名在项目的 config.json 文件中：

```
"module": {
  "package": "com.hellodemos.commevent"
}
```

然后单击“添加”按钮，在服务器上生成与真机相关的信息，如图 3-38 所示。



图 3-38 在服务器上生成真机

需要填写如下一些信息，如图 3-39 所示。

HarmonyAppProvision信息

包名： com.hellodemos.commonevent

* 名称： 真机测试 4/100

* 类型： 发布 调试

* 选择证书： 测试证书

选择

* 选择设备： 已选择1个

选择

申请受限权限： 已选择2个 选择

提交 取消

图 3-39 填写生成真机信息

- 名称是必填项，随便填写即可。
- 类型选择调试，因为我们只需要在真机上调试，不会到应用商店发布。
- 证书自己选择前面生成的证书，这里会弹出选择框，让我们选择。
- 设备选择我们前面加入的真机设备。
- 申请受限权限先全部申请，毕竟只是调试用。

提交之后，单击“下载”按钮，将 p7d 文件下载下来，如图 3-40 所示。



图 3-40 下载 p7d 文件

经过以上操作后，我们一共有 4 个证书文件。其中 huawei1.p12、huawei1.csr 是 DevEco-

Studio 开发环境中生成的，真机测试 Debug.p7b、测试证书是 AppGallery Connect 网站生成的，这些证书的目的在于保证安装在我们真机上的 App 是有认证的，安全的。4 个证书如图 3-41 所示。

名称
真机测试 Debug.p7b
测试证书.cer
huawei1.csr
huawei1.p12

图 3-41 4 个 App 认证证书

3.3.4 在开发环境中配置相关信息

最后一步，在 DevEco-Studio 中配置刚才的证书，就可以将 App 下载到真机上调试了。打开 File > Project Structure，在 Modules>entry（模块名称）>Signing Configs > debug 窗口中，配置指定模块的调试签名信息，如图 3-42 所示。

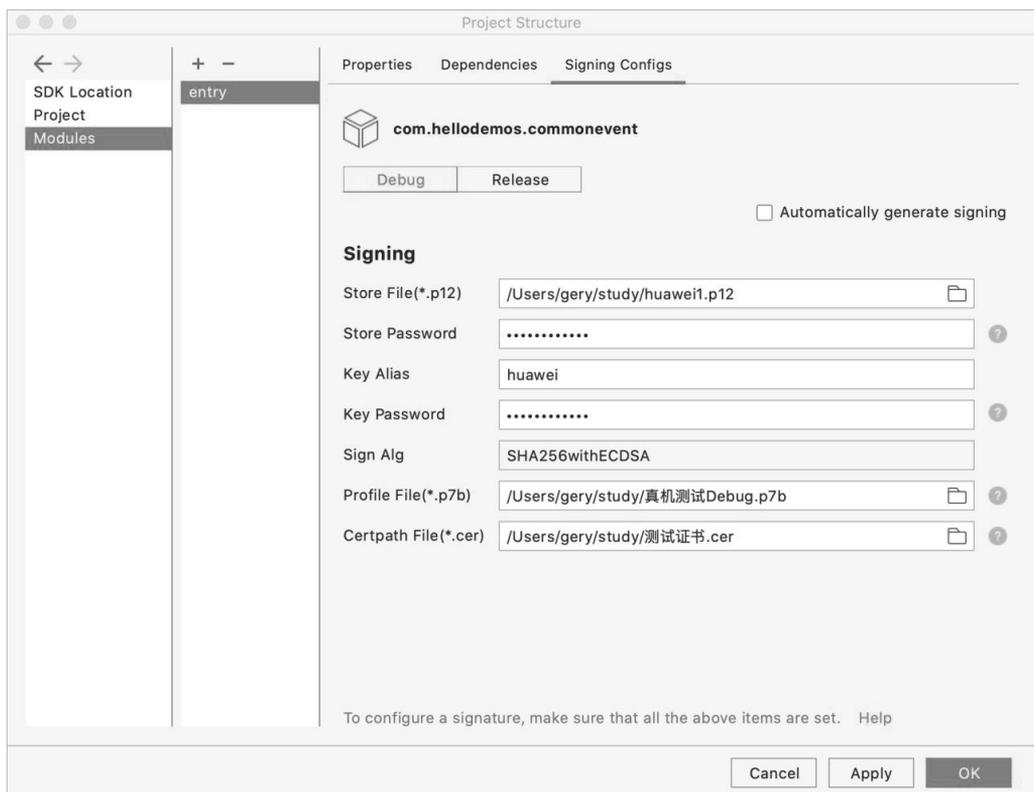


图 3-42 配置模块调试信息

- **Store File:** 选择密钥库文件，文件后缀为 `p12`。
- **Store Password:** 输入密钥库密码。
- **Key Alias:** 输入密钥的别名信息。
- **Key Password:** 输入密钥密码。
- **SignAlg:** 签名算法，固定为 `SHA256withECDSA`。
- **Profile File:** 选择申请的调试 `Profile` 文件，文件后缀为 `p7b`。
- **Certpath File:** 选择申请的调试数字证书文件，文件后缀为 `cer`。

3.3.5 运行程序

经过上面的若干步后，就可以将手机通过 `USB` 数据线连接上计算机，然后单击“运行”按钮，就可以将本章最开始的程序发送到真机上运行了，如图 3-43 所示。

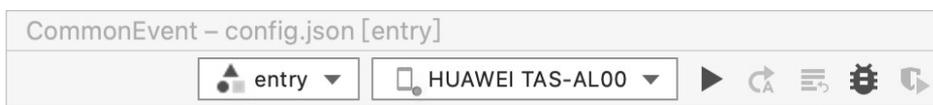


图 3-43 运行应用在测试真机上

3.4 小结

通过本章的学习，你一定会对怎样开发一个鸿蒙应用有了初步的了解，包括创建项目、申请应用开发者账号以及在远程模拟器及真机上运行程序，这些过程比较烦琐，需要我们自己实践一次，才能彻底掌握，大家快开始动手吧。开发一个 `Demo` 很容易，但是要开发出一个优秀的程序，任重而道远，接下来，我们还需要学习和掌握更多的知识。