

鸿蒙 OS C 语言设备开发基础

本章主要内容：

- (1) 鸿蒙操作系统 HarmonyOS。
- (2) 虚拟机管理软件 VMware Workstation。
- (3) Linux 操作系统(Ubuntu)虚拟机。
- (4) 鸿蒙 OS C 语言设备开发编译环境。
- (5) 华为 DevEco Device Tool。
- (6) 鸿蒙 OS C 语言设备开发集成开发环境。
- (7) Visual Studio Code。
- (8) 鸿蒙 OS C 语言设备开发案例：点亮一只 LED 灯。

本章对鸿蒙操作系统及其技术特征进行讲解，其中可能涉及一些对初学者来说比较陌生的专业术语，这是为了保证内容的完整性。在学习本章时如果遇到困难，并不会影响到后续 C 语言的学习和鸿蒙 OS C 语言设备开发实验的完成。

3.1 鸿蒙 OS 简介

鸿蒙操作系统 HarmonyOS(以下简称鸿蒙 OS)是中国华为技术有限公司在 2019 年 8 月 9 日发布的面向全场景的分布式操作系统，它利用分布式软总线技术将人、设备、场景有机地联系在一起，创造一个万物互联的世界，可帮助智能终端设备实现极速发现、极速连接、硬件互助、资源共享，用最合适的设备提供最佳的场景体验。

3.1.1 鸿蒙 OS 的发展历史

2012 年，华为公司开始规划开发自有操作系统鸿蒙 OS。

2019 年 5 月 17 日，由任正非领导的华为公司操作系统团队完成了自主知识产权操作系统鸿蒙 OS 的开发。

2019 年 8 月 9 日，华为公司正式发布鸿蒙 OS。

2020 年 9 月 10 日，华为公司鸿蒙 OS 升级至 2.0 版本，即 HarmonyOS 2.0，并面向 128KB~128MB 内存的终端设备开源。

2020 年 12 月 16 日，华为公司正式发布鸿蒙 OS 2.0 手机开发者 Beta 版本。2020 年已有美的、九阳、老板电器、海雀科技等公司生产的设备安装了鸿蒙 OS。

2021 年 2 月 22 日晚，华为公司正式宣布鸿蒙 OS 应用开发在线体验网站计划

于4月上线。

2021年3月,安装鸿蒙 OS 的物联网设备(手机、平板计算机、手表、智慧屏、音箱等智慧物联产品)有望达到3亿台,其中手机将超过2亿台,鸿蒙 OS 生态的市场份额有望达到16%。

2021年4月22日,华为公司鸿蒙 OS 应用开发在线体验网站上线。

2021年5月21日,华为公司宣布华为 HiLink 将与鸿蒙 OS 统一为鸿蒙 OS Connect。

2021年6月2日晚,华为公司正式发布 HarmonyOS 2.0 及多款搭载 HarmonyOS 2.0 的新产品。7月29日,华为公司 Sound X 音箱发布,是首款搭载 HarmonyOS 2.0 的智能音箱。

2021年10月,华为公司宣布搭载鸿蒙设备突破1.5亿台。鸿蒙 OS 座舱汽车于2021年底发布。

2021年11月17日,鸿蒙 OS 迎来第三批开源,新增开源组件769个,涉及工具、网络、文件数据、UI、框架、动画图形及音视频7大类。

2023年8月4日,华为公司 HarmonyOS 4.0 操作系统正式发布。

2023年8月4日,华为公司 HarmonyOS Next 操作系统开发者预览版(Developer Preview)正式发布。

2024年1月10日,华为公司宣布与网易游戏达成合作,网易首款鸿蒙原生手游《倩女幽魂》完成开发。同日,京东启动鸿蒙原生应用开发。

3.1.2 鸿蒙 OS 的设计理念

鸿蒙 OS 的设计理念是实现万物智能互联,依托基于网络通信的软总线技术,将人类生产和生活中孤立的、功能相对单一的各种各样的设备有机联系起来,实现不同设备的资源融合、资源共享;通过智能化的设备管理、任务管理、数据处理,针对不同类型的业务,为业务匹配和选择最恰当的执行硬件,让业务在不同设备间按设备能力匹配流转和处理,充分发挥不同设备各自的能力优势,如摄像能力、显示能力、传感能力、控制能力、交互能力、数据处理能力等;针对不同设备在屏幕尺寸、交互方式、使用场景、用户人群等方面的存在的差异性进行专门的差异性设计,从而给用户提供最合适的使用体验;针对纷繁复杂的设备和使用场景,尽可能在界面设计和交互方式上保持一致性和人性化,使用户方便学习、易于使用,从而创造一个智慧化的万物互联的世界。

鸿蒙 OS 系统功能设计按照系统→子系统→功能→模块逐级展开。可根据实际需求裁剪某些非必要的子系统或功能/模块。

3.1.3 鸿蒙 OS 的技术架构

鸿蒙 OS 的技术架构如图 3-1 所示,系统采用分层设计,从下往上分别为内核层、系统服务层、框架层和应用层。各层组成及功能如下。

1. 内核层

内核层主要由内核子系统和驱动子系统构成。

(1) 内核子系统。鸿蒙 OS 采用多内核设计,针对不同的设备会选用与其匹配的操作系统内核。为了屏蔽多内核差异对上层的影响,内核子系统设计采用内核抽象层(Kernel Abstract Layer, KAL)实现对上层一致的进程/线程管理、内存管理、文件系统管理、网络管



鸿蒙 OS 的设计理念



鸿蒙 OS 的技术架构

理和外设管理。

(2) 驱动子系统。由硬件驱动框架(Hardware Driven Frame, HDF)构成,提供统一的外设访问能力、驱动开发及管理框架,是鸿蒙 OS 对不同种类的硬件提供支持的基础。

2. 系统服务层

系统服务层是鸿蒙 OS 的核心,它通过框架层对应用层的程序提供服务。该层由以下几部分组成:

(1) 系统基本能力子系统集。由分布式任务调度、分布式数据管理、分布式软总线、方舟多语言运行时、公共基础库、多模输入、图形、安全和人工智能等子系统组成,为分布式应用业务在多设备上的运行、调度、迁移等操作提供基础支撑。其中,方舟多语言运行时不但为 C、C++、JavaScript 等多语言运行提供支持和基础的系统类库,而且也在使用方舟编译器静态化编译的 Java 程序(即应用程序或框架层中使用 Java 语言开发的部分)提供运行支持。

(2) 基础软件服务子系统集。由事件通知、电话、多媒体、DFX(面向产品生命周期各环节的设计)、MSDP&DV(多播源发现协议和数字视频)等子系统组成,使鸿蒙 OS 能提供公共的、通用的软件服务。

(3) 增强软件服务子系统集。由智慧屏专有业务、穿戴专有业务、IoT 专有业务等子系统组成,使鸿蒙 OS 能提供针对不同设备的差异化的能力增强型软件服务。

(4) 硬件服务子系统集。由位置服务、生物特征识别、穿戴专有硬件服务、IoT 专有硬件服务等子系统组成,为鸿蒙 OS 提供硬件服务。

鸿蒙 OS 能根据不同设备形态配置不同的部署环境,可根据需要对基础软件服务子系统集、增强软件服务子系统集、硬件服务子系统集进行裁剪,每个子系统内部又可以按业务功能粒度裁剪。

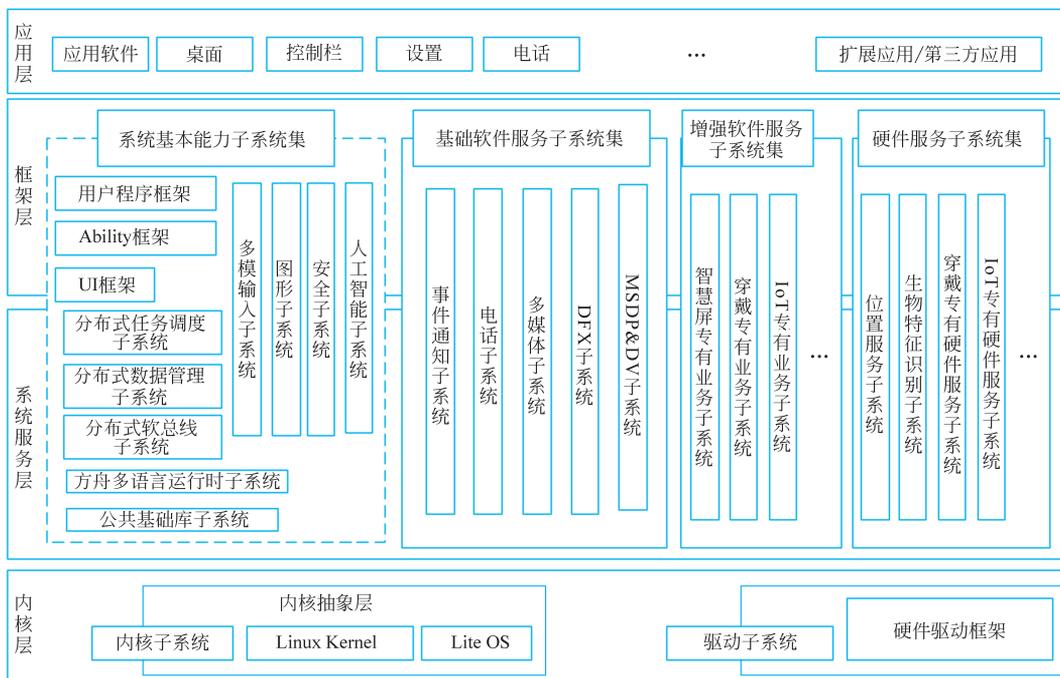


图 3-1 鸿蒙 OS 的技术架构

3. 框架层

框架层为鸿蒙 OS 应用开发提供 Java、C、C++、JavaScript 等多语言的用户程序框架、Ability 框架、两种 UI 框架(包括适用于 Java 语言的 Java UI 框架和适用于 JavaScript 语言的 JavaScript UI 框架)以及各种软硬件服务对外开放的多语言框架 API,根据系统的组件化裁剪程度,鸿蒙 OS 设备支持的 API 也会有所不同。

4. 应用层

应用层包括系统应用和第三方非系统应用。鸿蒙 OS 的应用由一个或多个 FA (Feature Ability)或 PA(Particle Ability)组成。其中,FA 有用户界面,提供与用户交互的能力;而 PA 没有用户界面,提供后台运行任务的能力以及统一的数据访问抽象。FA 在进行用户交互时所需的后台数据访问也需要由对应的 PA 提供支撑。基于 FA/PA 开发的应用能够实现特定的业务功能,支持跨设备调度与分发,为用户提供一致、高效的应用体验。

3.1.4 鸿蒙 OS 的技术特点

依托分布式软总线、分布式设备虚拟化、分布式数据管理、分布式任务调度等关键技术,鸿蒙 OS 支持不同种类设备之间的硬件互助、资源共享,具备分布式软总线、分布式数据管理和分布式安全三大核心能力。

1. 分布式软总线

计算机硬件系统通过地址总线、数据总线和控制总线,在计算机的运算器、存储器、控制器和输入输出设备之间传递地址、数据和控制信息,以实现计算机的功能。华为公司借鉴这种硬件总线设计思想,依托其业界领先的信息传输技术,利用鸿蒙 OS 在手机、平板计算机、智能穿戴设备、智慧屏、车载电子设备、音响、空调等分布式设备之间打造了如图 3-2 所示的基于网络传输的分布式软总线架构,为不同种类设备之间的互联互通提供了统一的分布式通信能力,提供高带宽、低时延、高可靠和安全的数据传输通道,为设备之间的智能感知发现和零等待传输创造条件,使开发者只需要聚焦于业务逻辑的实现,而无须关注组网方式与底层协议。

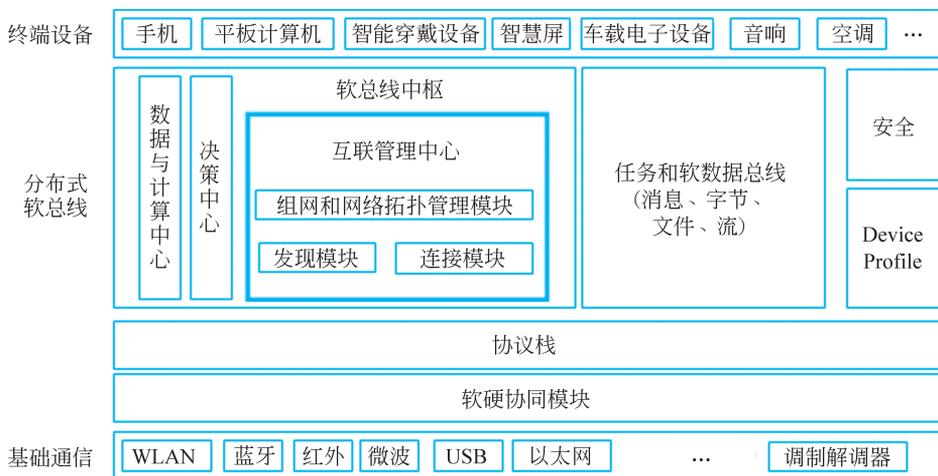


图 3-2 鸿蒙 OS 分布式软总线架构

利用这种数据软总线技术,可以轻松实现手机与家用电器设备的智能互联。例如,无须烦琐的设置,只需通过碰一碰就可以实现手机与微波炉、电烤箱、电灯、电视、空调、空气净化



鸿蒙 OS 的
技术特点

器、油烟机、加湿器、音响等家用电器的即连即用,控制这些设备完成其各自的任务;又如,根据自己的喜好播放预先设定的音乐或者电视节目,或者自动按照设定的菜谱和烹调参数烹制自己喜欢的美味佳肴。

2. 分布式设备虚拟化

分布式设备虚拟化是在鸿蒙 OS 的支持下,将功能单一的不同设备虚拟化成功能多样的一体设备,然后根据具体业务的功能实现需求,自动调用和管理适合处理这种业务的设备资源,使其进行高效的数据处理。也就是说,针对不同类型的任务,鸿蒙 OS 可以为其选择能力合适的执行硬件,使业务在不同设备间按处理需求自动流转,充分发挥不同设备的能力优势,如显示能力、摄像能力、音频能力、交互能力以及传感器能力等。例如,用户可以边做家务边接听视频电话,此时会将手机自动与智慧屏、摄像头、音箱连接,并将其虚拟化为手机资源,替代手机自身的屏幕、摄像头、听筒,实现一边做家务一边通过智慧屏、摄像头、音箱进行视频通话。

3. 分布式数据管理

鸿蒙 OS 可以利用分布式软总线技术对应用程序和用户数据实现分布式管理,使特定数据既不再与特定软件绑定处理,也不再与特定设备绑定存储。例如,正在摄录的视频数据不一定就存储在正在摄录的摄像设备上,正在播放的音乐也不一定就存储在正在播放的音响上,可以将其存储在适合存储它们的大容量存储器上。鸿蒙 OS 还可以使业务处理逻辑与数据存储分离。例如,计算机正在处理的数据不一定存储在自己的存储器上,可能来自其他任何设备。鸿蒙 OS 也可以使跨设备的数据处理如同处理本地数据一样方便快捷,使开发者、用户能够轻松实现全场景、多设备下的数据分布式存储、共享和访问。

例如,可以将手机上的文档自动传送到平板计算机,在平板计算机上对文档进行编辑、查阅、删除等操作,文档的最新状态可以在手机上同步显示。

4. 分布式任务调度

分布式任务调度是鸿蒙 OS 利用分布式软总线、分布式数据管理、分布式 Profile 等技术提供的分布式设备服务统一管理(发现、同步、注册、调用)机制,支持应用进程在不同设备之间的远程启动、远程调用、远程连接以及迁移等操作。任务分配与调度根据不同设备的能力、位置、业务运行状态、资源使用情况以及用户的习惯和意图进行精准分配。

例如,用户驾车出行时,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载导航设备,下车后导航自动迁移回手机;用户骑车出行,在手机上规划好导航路线,骑行时手表可以接续导航;在手机上点外卖后,可以将订单信息迁移到手表上,随时查看外卖的配送状态。

5. 一次开发,多种设备部署

鸿蒙 OS 提供用户程序框架、Ability 框架以及 UI 框架,支持多种设备终端业务逻辑和界面逻辑的复用,能够实现应用程序的一次开发、多种设备部署,提升了跨设备应用的开发效率。

其中,UI 框架支持 Java 和 JavaScript 两种开发语言,并提供丰富的多态控件,可以在手机、平板计算机、智能穿戴设备、智慧屏、车机上显示不同的用户界面效果。用户界面采用业界主流设计方式,提供多种自适应式布局方案,支持栅格化布局,具备不同屏幕的界面自动匹配能力。

6. 系统统一,弹性部署

鸿蒙 OS 采用组件化设计方法,根据不同终端设备的硬件资源和功能需求,按需安装鸿蒙 OS 组件;支持通过编译链关系自动生成组件化的依赖关系,形成组件树依赖图;支持产品系统的便捷开发,降低硬件设备的开发门槛。

3.1.5 鸿蒙 OS 的应用场景

目前,尽管我国的信息技术应用已经普及到各个领域,但“缺心少魂”现象非常严峻。在手机、计算机等信息技术硬件方面,关键芯片受制于人;在操作系统方面,Windows 系列操作系统、Linux 操作系统、macOS 操作系统等一直垄断着计算机领域,安卓系统、iOS 操作系统一直霸占着手机领域。近期频发的“卡脖子”现象严重制约着我国信息技术产业的健康发展,危及信息安全,给个人隐私保护、社会稳定、国家安全埋下严重的隐患。在此关键时刻,“混沌初开、鸿蒙出世”,华为公司攻坚克难,终于在 2019 年发布了具有中国自主知识产权的鸿蒙 OS。鸿蒙 OS 虽因华为手机而生,但并不仅仅是代替安卓系统应用于华为手机,它是为万物互联而诞生的,可以应用于除手机以外的很多系统。

1. “1+8+N”的多场景战略

与应用于单一领域的安卓手机操作系统、苹果手机操作系统、Linux 计算机操作系统不同,鸿蒙 OS 的应用秉持“1+8+N”战略。其中,“1”是指以华为手机和用户为中心和起点;“8”是指鸿蒙 OS 可以应用于 8 种常用的电子设备,包括计算机、大屏幕、空调、音响、平板计算机、手表/手环、车载电子设备、AR/VR 设备;“N”是指万物互联,也就是物联网。在鸿蒙 OS 支持下,能够实现智能家居、运动健康、影音娱乐、智慧出行、移动办公、生产制造等众多领域电子设备的智能互联。

2. 面向人工智能的操作系统

鸿蒙 OS 融合人工智能和物联网两种技术,目的是打造电子产品和机械产品的智能化和互联互通,任何电子产品和机械产品只要安装了它,就会变成智能化的硬件,相互间很容易实现数据共享、任务互助。例如,可以将手机的导航信息显示到手表上,并且以图示和语音信息进行导航提示,这在人们双手提着行李不方便看手机时就非常有用;又如,将手机的视频信息自动映射到大屏幕上,就可以有更好的视觉享受。这样就可以打破设备壁垒,扩展设备能力,实现多端互助。

3. 与安卓生态兼容并蓄,无缝对接

鸿蒙 OS 兼容安卓手机操作系统,包含安卓系统拥有的功能,非常便于安卓生态资源的迁移。如同在平板计算机或者计算机上可以同时安装多个操作系统一样,当要使用某个操作系统时,用户就可以切换到那个操作系统并进行相应的操作。例如,用户在计算机上安装了 Windows 和 Linux 双系统。如果当前正在使用 Linux,而用户要使用某个 Windows 系统的软件,这个时候用户只要利用某种方法切换到 Windows 系统即可。类似地,借助仿真器(EMU),华为手机也可以非常方便地从安卓系统无缝地切换为鸿蒙 OS。

4. 传承过去,引领未来

鸿蒙 OS 传承过去,融合传统操作系统的已有特点,同时引领未来,朝着万物智能互联不断地发展突破。



鸿蒙 OS 的应用场景

3.2 鸿蒙 OS C 语言设备开发实验板

与本书配套的鸿蒙 OS C 语言设备开发实验是利用 Bossay 鸿蒙 OS C 语言设备开发实验板进行的,这个开发实验板如图 3-3 所示,它包括一个集成了 Hi3861 模组的核心底板、一个集成了



鸿蒙 OS C 语言设备开发实验板

光照强度传感器的炫彩灯案例板、一个数码管点阵板实验案例板和一个 NFC 电子标签等。

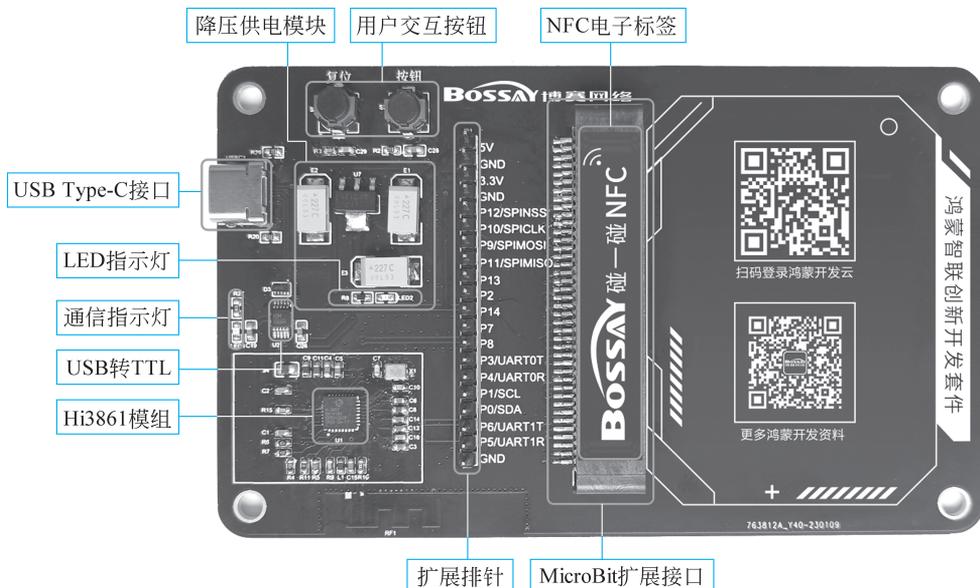


图 3-3 Bossay 鸿蒙 OS C 语言程序设备开发实验板

核心底板是 Hi3861 WLAN 模组,它是一片 115mm×72mm 大小的开发板。核心底板主要包括以下部件: Hi3861 模组、CH340USB 转串口芯片、USB Type-C 接口、复位按钮、可编程的自定义按钮、LED 指示灯、MicroBit 案例板扩展接口、杜邦线扩展排针等,具体功能描述如下:

- Hi3861 芯片内置的 Flash 用于存放二进制文件代码与配置参数等静态数据,内置的 CPU 用于执行程序,内置的 SRAM 用于保存程序运行时的数据,内置的 WiFi 功能为应用程序提供网络连接的能力。
- CH340USB 转串口芯片和 USB Type-C 接口是用于开发实验板和 PC 连接、固件烧录、运行调试的外设接口。
- 用户交互按钮中的复位按钮用于整板复位;另一个按钮用于自定义功能,可以通过用户程序进行功能控制或触发中断。
- 板载的两个 LED 指示灯位于 USB Type-C 接口附近,标号 LED2 的指示灯用于整板供电指示,标号 LED1 的指示灯通过闪烁指示和 PC 的通信数据传输。
- MicroBit 案例板扩展接口用于扩展丰富的外设案例板,对标行业应用。除与本书配套的案例板外,如果学生有创新需求,可以登录博赛网络案例板商城购买或提出定制需求。
- 板载的 20 针杜邦线扩展排针方便用户扩展非标准接口的外设传感器模组或执行器模组。
- NFC 电子标签粘贴在 MicroBit 插槽上。用户用其写入自定义标签数据后,可通过鸿蒙 OS 碰一碰功能调取指定的手机端 App,并完成无感 WiFi 配网。

板载的 Hi3861 WLAN 模组是一款高度集成的 2.4GHz WLAN SoC 芯片,其芯片外设接口如图 3-4 所示,表 3-1 列出了 Hi3861 芯片外设接口的详细信息。

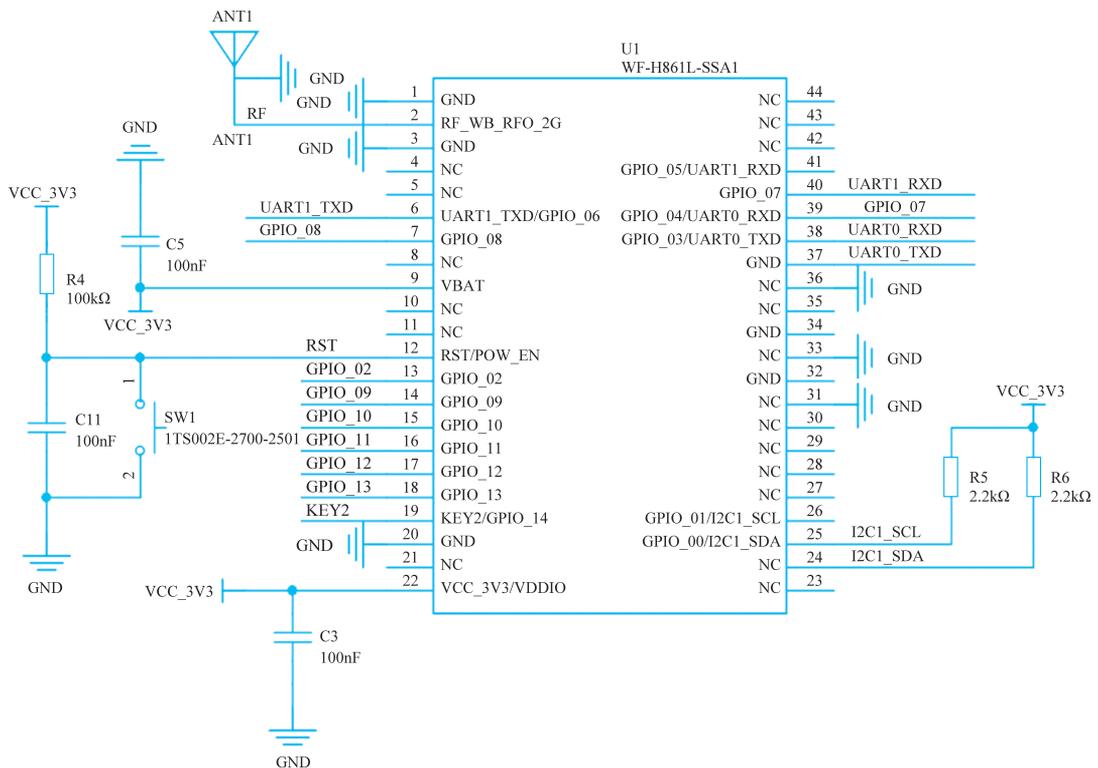


图 3-4 Hi3861 芯片外设接口

表 3-1 Hi3861 芯片外设接口

序号	名称	功能	序号	名称	功能
1	GND	接地	15	GPIO_10	输入输出
2	RF_WB_RFO_2G	射频接口	16	GPIO_11	输入输出
3	GND	接地	17	GPIO_12	输入输出
4	NC	空连接	18	GPIO_13	输入输出
5	NC	空连接	19	KEY2/GPIO_14	按键
6	UART1_TXD/GPIO_06	异步传输	20	GND	接地
7	GPIO_08	输入输出	21	NC	空连接
8	NC	空连接	22	VCC_3V3/VDDIO	电源
9	VBAT	电池工作	23	NC	空连接
10	NC	空连接	24	NC	空连接
11	NC	空连接	25	GPIO00/I2CI_SDA	输入输出
12	RST/POW_EN	复位	26	GPIO01/I2CI_SCL	输入输出
13	GPIO_02	输入输出	27	NC	空连接
14	GPIO_09	输入输出	28	NC	空连接

续表

序号	名称	功能	序号	名称	功能
29	NC	空连接	37	GND	接地
30	NC	空连接	38	GPIO_03/UART0_TXD	输入输出
31	NC	空连接	39	GPIO_04/UART0_RXD	输入输出
32	GND	接地	40	GPIO_07	输入输出
33	NC	空连接	41	GPIO_05/UART1_RXD	输入输出
34	GND	接地	42	NC	空连接
35	NC	空连接	43	NC	空连接
36	NC	空连接	44	NC	空连接

该芯片集成 IEEE 802.11b/g/n 基带和 RF (Radio Frequency, 射频) 电路。支持 HarmonyOS, 通过了 HarmonyOS Connect 认证, 并配套提供开放、易用的开发和调试运行环境。其中 Hi3861V100 MCU (MicroController Unit, 微控制单元) 是海思半导体公司生产的一款高度集成的 2.4GHz SoC WiFi 芯片, 集成 IEEE 802.11b/g/n 基带和 RF 电路, RF 电路包括功率放大器、低噪声放大器、射频巴伦变压器、天线开关以及电源管理等模块; 支持 20MHz 标准带宽和 5MHz/10MHz 窄带宽, 提供最大 72.2Mb/s 物理层速率。Hi3861V100 WiFi 基带支持正交频分复用 (Orthogonal Frequency Division Multiplexing, OFDM) 技术, 并向下兼容直接序列扩频 (Direct Sequence Spread Spectrum, DSSS) 和补码键控 (Complementary Code Keying, CCK) 技术, 支持 IEEE 802.11 b/g/n 协议的各种数据速率。Hi3861V100 芯片集成高性能 32 位微处理器、硬件安全引擎以及丰富的外设接口, 外设接口包括 SPI (Serial Peripheral Interface, 串行外设接口)、UART、I2C (Philips 公司开发的双向二线制同步串行总线)、PWM (Pulse Width Modulation, 脉冲宽度调制)、GPIO (General Purpose Input/Output, 通用输入输出) 和多路 ADC (Analog to Digital Converter, 模数转换器), 同时支持高速 SDIO2.0 Slave 接口, 最高时钟频率可达 50MHz; Hi3861V100 芯片内置 SRAM 和 Flash, 可独立运行, 并支持在 Flash 上运行程序。Hi3861V100 支持华为 Lite OS 和第三方组件, 并配套提供开放、易用的开发和调试运行环境。Hi3861V100 芯片适用于智能家电等物联网智能终端领域。

3.3 鸿蒙 OS C 语言设备开发环境

注意: 登录清华大学出版社官方网站或济南博赛网络技术有限公司的官方网站, 完成注册并登录后, 可以获得与本书配套的搭建鸿蒙 OS C 语言设备开发环境所需的有关软件以及配套的软件安装配置文档和软件安装教学视频。

工欲善其事, 必先利其器。要想利用 C 语言进行基于鸿蒙 OS 的应用开发, 首先必须掌握鸿蒙 OS C 语言设备开发环境。鸿蒙 OS C 语言设备开发环境如图 3-5 所示, 由硬件环境和软件环境组成。

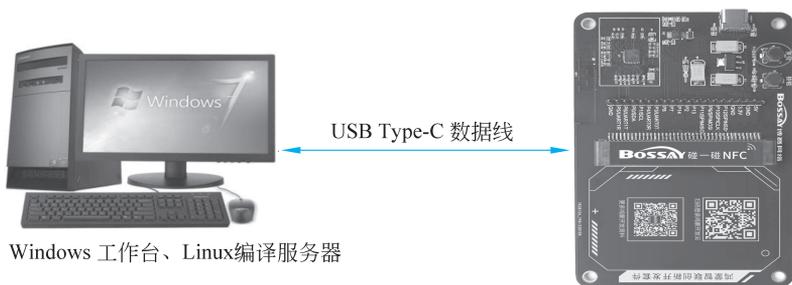


图 3-5 鸿蒙 OS C 语言设备开发环境

3.3.1 鸿蒙 OS C 语言设备开发硬件环境

鸿蒙 OS C 语言设备开发硬件环境由下列 4 部分组成。

1. Windows 工作台

Windows 工作台由一台安装了 Windows 操作系统(Windows 10 及以上版本)的计算机组成。其作用如下:

- (1) 编辑 C 语言源程序代码。
- (2) 烧录二进制文件,也就是将编译好的可执行程序(二进制目标代码)写入支持鸿蒙操作系统的硬件设备(如博赛开发实验板、智能电视、智能空调、智能音响等)。

2. Linux 编译服务器

Linux 编译服务器既可以在 Windows 工作台上安装的 Linux 虚拟机(安装 Ubuntu 版本的操作系统)组成,也可以由一台安装了 Linux 操作系统(Ubuntu 版本)的独立计算机组成。它主要用于编译 C 语言源程序代码,将使用 Windows 工作台编辑的 C 语言源程序代码编译成可以在鸿蒙 OS 支持的硬件设备上运行的 C 语言可执行程序(二进制目标代码)。

3. 鸿蒙 OS C 语言开发实验设备

鸿蒙 OS C 语言开发实验设备是由济南博赛网络技术有限公司研发的 Bossay 实验套件或实验箱。

4. USB Type-C 数据线

使用 USB Type-C 数据线将 Windows 工作台和鸿蒙 OS C 语言开发实验板连接起来,将可执行代码从 Windows 工作台烧录到鸿蒙 OS C 语言开发实验板上。

3.3.2 鸿蒙 OS C 语言设备开发软件环境

鸿蒙 OS C 语言设备开发软件环境如表 3-2 所示。

表 3-2 鸿蒙 OS C 语言设备开发软件环境

平台	需要安装的软件	软件安装程序或软件版本	说明
Windows 工作台	Windows 操作系统	Windows 10 及以上	
	Visual Studio Code	Visual Studio Code 1.66.2	C 语言源代码编辑工具
	DevEco Device Tool Beta1	devicetool-windows-tool-3.1.0.300.zip	Windows 版鸿蒙 OS C 语言程序集成开发工具软件



鸿蒙 OS C
语言设备开发
硬件环境



鸿蒙 OS C
语言设备开发
软件环境

续表

平台	需要安装的软件	软件安装程序或软件版本	说 明
Windows 工作台	Python	Python 3.8.10	C 语言源代码编译依赖工具
	烧录软件 HiBurn	HiBurn.exe	V2.4 及以上版本
Linux 编译服务器	虚拟机管理软件 VMware Workstation	Oracle VM VirtualBox 16.2.3 及以上版本	与本书配套的是 VMware-workstation- full-16.2.3-19376536.exe
	Linux 操作系统软件 Ubuntu 20.04	ubuntu-20.04.2.0-desktop- amd64.iso	Shell 使用 bash
	DevEco Device Tool Beta1	devicetool-linux-tool-3.1.0. 300.zip	Linux 版鸿蒙 OS C 语言程序集成开发 工具软件,该软件是 Python、gn、ninja、 gcc_riscv32、Scons、Node.js、hpm 等软件 的集合
	安装包管理工具 apt	与 Ubuntu 20.04 配套版本	在线网络下载安装
	网络管理软件 net-tools	与 Ubuntu 20.04 配套版本	在线网络下载安装
	开源文件传输工具软件 curl	与 Ubuntu 20.04 配套版本	在线网络下载安装
	开源的分布式版本控制 系统 git	与 Ubuntu 20.04 配套版本	在线网络下载安装
	vim 编辑工具	与 Ubuntu 20.04 配套版本	在线网络下载安装
	文件共享服务器 Samba	与 Ubuntu 20.04 配套版本	Samba 是在 Linux 和 UNIX 系统上实现 SMB 协议的一个免费软件,为局域网内 的不同计算机之间提供文件及打印机等 资源的共享服务
	远程登录的服务器 openssh-server	与 Ubuntu 20.04 配套版本	提供远程登录服务



安装配置鸿蒙 OS C 语言设备开发的编译环境

3.4 安装配置鸿蒙 OS C 语言设备开发的编译环境

在 Windows 工作台上安装配置鸿蒙 OS C 语言设备开发的编译环境,也就是安装编译 C 语言程序的下列软件,并配置软件运行环境。

- (1) 安装 Windows 操作系统(最好是 Windows 10 以上版本)。
- (2) 安装虚拟机管理软件 VMware Workstation。
- (3) 安装 Linux(Ubuntu 版本)操作系统的虚拟机。

(4) 在安装 Linux 操作系统的虚拟机上,安装配置鸿蒙 OS C 语言设备开发环境所需的 DevEco Device Tool 3.1 Beta1 Linux(编译服务端)软件。DevEco Device Tool 3.1 Beta1 Linux(编译服务端)软件由配置鸿蒙 OS C 语言程序编译环境所需的 Python、gn、ninja、gcc_riscv32、Scons、Node.js、hpm 等软件组成。

安装虚拟机
管理软件
VMware
Workstation

3.4.1 安装虚拟机管理软件 VMware Workstation

安装虚拟机管理软件 VMware Workstation 的方法和步骤如下：

(1) 如图 3-6 所示,在计算机硬盘上找到已下载的 VMware Workstation 的安装程序,本书使用的安装程序是 VMware-workstation-full-16.2.3-19376536.exe,存放在计算机的 D:\HarmonyOS C SETUP\Ubuntu Server Setup 文件夹下。读者可以从虚拟机管理软件 VMware Workstation 的官方网站下载该程序到自己的计算机,也可以从与本书配套的网站下载该程序。双击该程序,程序开始执行,弹出如图 3-7 所示的“VMware 产品安装”对话框,稍等片刻,接着出现如图 3-8 所示的 VMware Workstation Pro 安装向导。



图 3-6 VMware Workstation 安装程序



图 3-7 “VMware 产品安装”对话框



图 3-8 VMware Workstation Pro 安装向导

(2) 单击图 3-8 所示对话框右下角的“下一步”按钮,出现如图 3-9 所示的“VMware 最终用户许可协议”对话框。

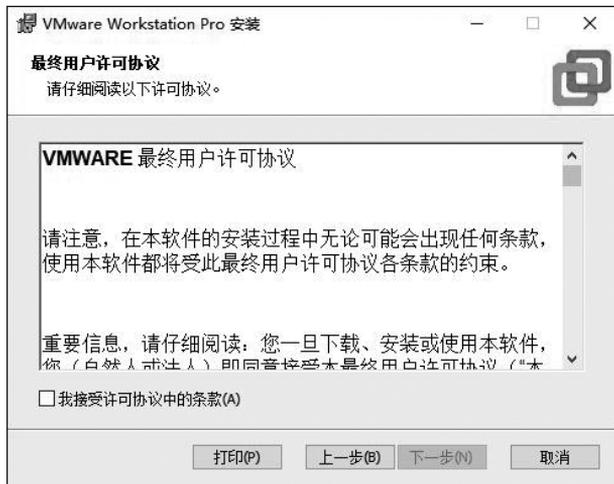


图 3-9 VMware 最终用户许可协议

(3) 选择图 3-9 所示对话框左下角的“我接受许可协议中的条款”复选框,此时对话框中右下角的“下一步”按钮由虚变实,单击该按钮,出现如图 3-10 所示的“自定义安装”对话框。

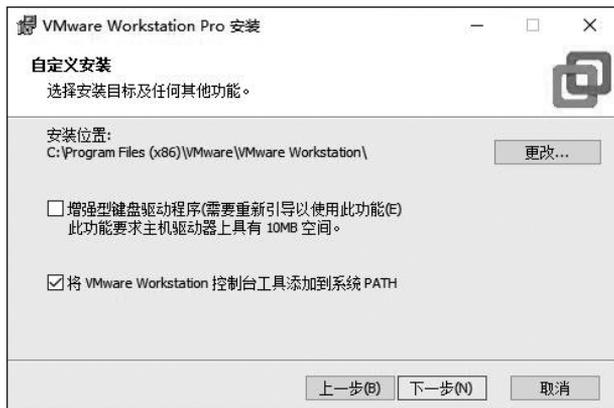


图 3-10 “自定义安装”对话框

(4) 在如图 3-10 所示的对话框中,软件的安装位置默认为 C:\Program Files (x86)\VMware\VMware Workstation\,此时可以单击对话框中的“更改”按钮更改安装位置,也可以保持此安装位置不变。设置好安装位置后,单击对话框右下角的“下一步”按钮,出现如图 3-11 所示的“用户体验设置”对话框。为避免安装后的虚拟机每次启动时都更新,取消对对话框中部的“启动时检查产品更新”和“加入 VMware 客户体验提升计划”两个复选框的选择,此时对话框如图 3-12 所示。

(5) 单击图 3-12 所示对话框右下角的“下一步”按钮,出现如图 3-13 所示的“快捷方式”对话框。选择该对话框中部的“桌面”和“开始菜单程序文件夹”两个复选框,此时对话框如图 3-13 所示。

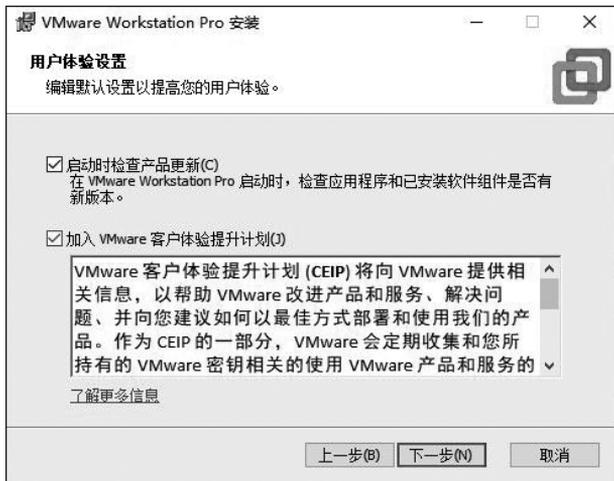


图 3-11 “用户体验设置”对话框

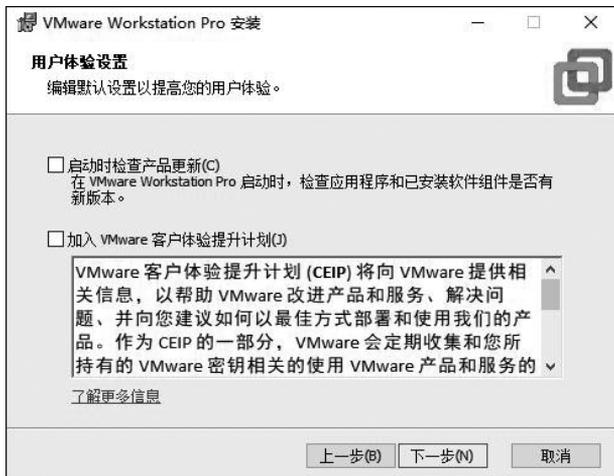


图 3-12 取消对两个复选框的选择



图 3-13 “快捷方式”对话框

(6) 单击图 3-13 所示对话框右下角的“下一步”按钮,出现如图 3-14 所示的“已准备好安装 VMware Workstation Pro”对话框。单击该对话框右下角的“安装”按钮,出现如图 3-15 所示的“正在安装 VMware Workstation Pro”对话框,表示安装程序正在将程序文件复制到计算机上并完成程序配置。等安装完成后出现如图 3-16 所示的“VMware Workstation Pro 安装向导已完成”对话框。在该对话框中可以单击“许可证”按钮,在弹出的对话框中输入许可证密钥,完成正版认证。也可以单击该对话框右下角的“完成”按钮,先结束程序安装。安装完成后,在 Windows 10 操作系统桌面上会出现如图 3-17 所示的 VMware Workstation Pro 快捷方式图标。

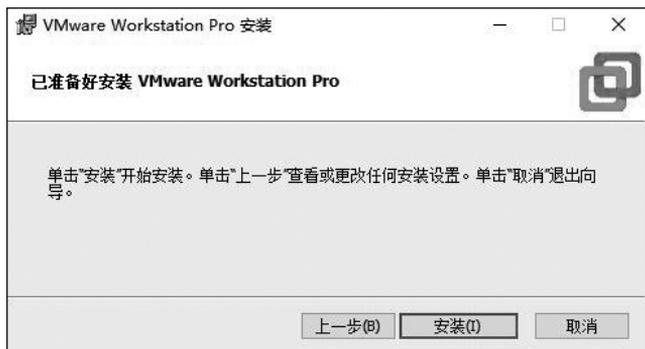


图 3-14 “已准备好安装 VMware Workstation Pro”对话框

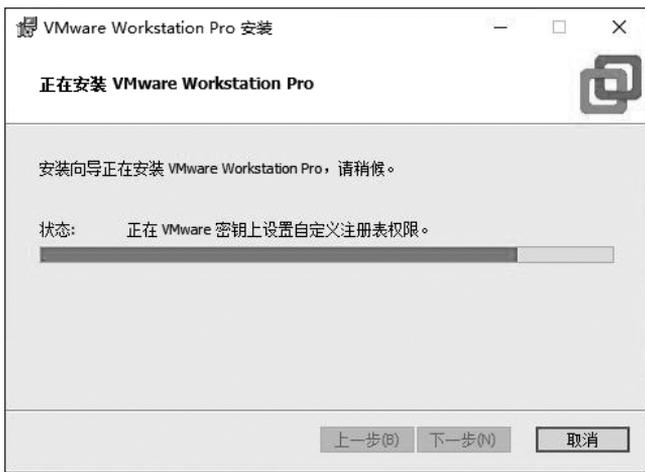


图 3-15 “正在安装 VMware Workstation Pro”对话框

(7) 如果第(6)步操作没有输入 VMware Workstation Pro 程序的许可证密钥,程序即使安装了,也只是试用版,试用时间有限。要想正常使用,还需要输入许可证密钥,方法是在 Windows 操作系统桌面上找到如图 3-17 所示的 VMware Workstation Pro 快捷方式图标,双击该图标启动 VMware Workstation Pro 程序,然后,如图 3-18 所示,在程序窗口中打开“帮助”菜单,选择“输入许可证密钥”命令,弹出如图 3-19 所示的“输入许可证密钥”对话框,在此输入由 25 个字符组成的许可证密钥,然后单击“确定”按钮,程序即可获得认证,成为正式版程序,就可以永久使用了。



图 3-16 “VMware Workstation Pro 安装向导已完成”对话框



图 3-17 VMware Workstation Pro 快捷方式图标

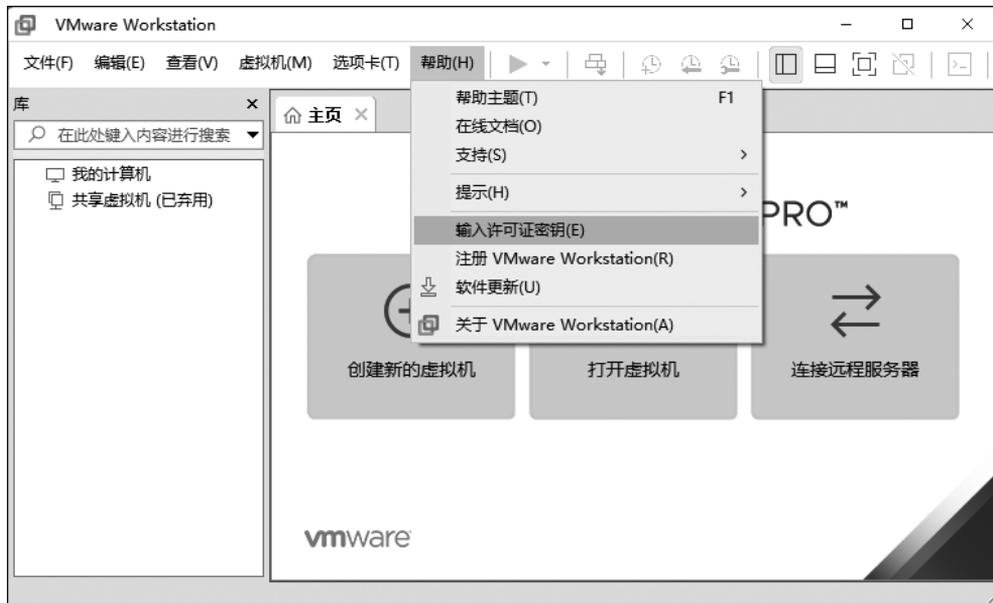


图 3-18 VMware Workstation Pro 程序窗口

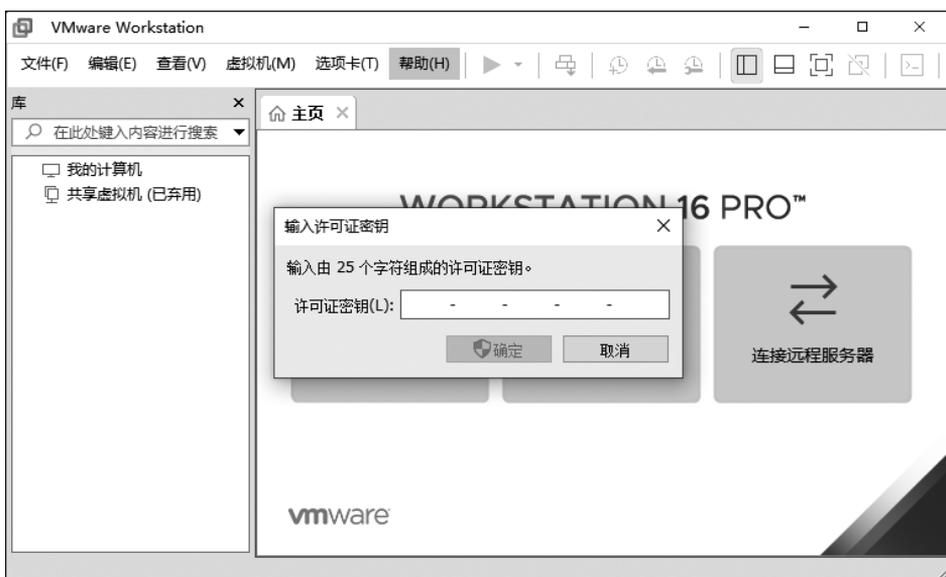


图 3-19 输入许可证密钥

3.4.2 使用 VMware Workstation 创建 Linux 虚拟机

注意：创建安装 Linux(Ubuntu 版本)操作系统的虚拟机时,因为需要从网站下载内容更新 Ubuntu,所以要确保计算机联通互联网,处于正常上网状态。

使用 VMware Workstation 软件创建安装 Linux(Ubuntu 版本)操作系统的虚拟机的步骤如下:

(1) 在计算机 Windows 操作系统桌面上,找到如图 3-17 所示的虚拟机程序 VMware Workstation Pro 的快捷方式图标,双击它,运行 VMware Workstation 程序,出现如图 3-20 所示的虚拟机程序 VMware Workstation 主窗口。

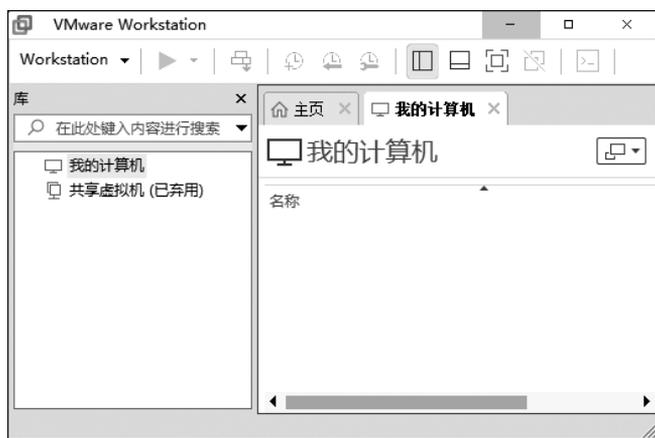


图 3-20 VMware Workstation 主窗口

(2) 在虚拟机程序 VMware Workstation 主窗口中,单击该窗口左上角的 Workstation,会出现如图 3-21 所示的菜单,在菜单中选择“文件”→“新建虚拟机”命令,出现如图 3-22 所示的



创建安装
Linux
(Ubuntu
版本)
虚拟机

新建虚拟机向导。

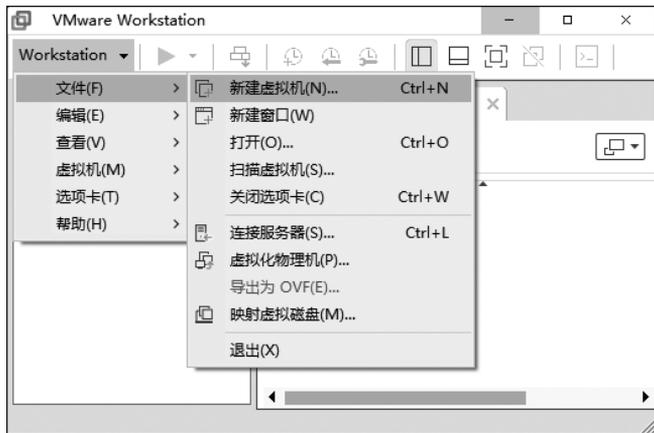


图 3-21 “新建虚拟机”命令

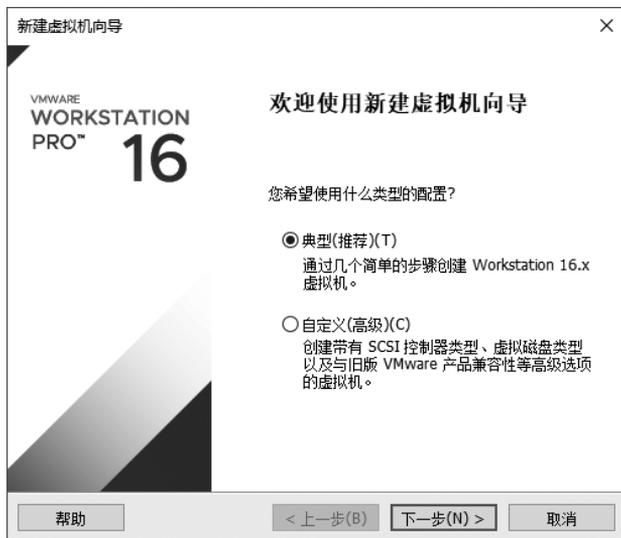


图 3-22 新建虚拟机向导

(3) 在新建虚拟机向导中,保持“典型(推荐)”单选按钮被选中,使用典型安装方式安装虚拟机,单击“下一步”按钮,出现如图 3-23 所示的“安装客户机操作系统”对话框。

(4) 在“安装客户机操作系统”对话框中,选择“安装程序光盘映像文件(iso)”单选按钮,然后单击其右侧的“浏览”按钮,出现如图 3-24 所示的“浏览 ISO 映像”对话框,目的是找到在计算机硬盘上保存的提前下载的 Linux 操作系统 Ubuntu 版本的安装映像文件。

注意: 本书使用的 Linux 操作系统 Ubuntu 版本的安装映像文件是 ubuntu-20.04.2.0-desktop-amd64.iso,该文件保存于 D:\HarmonyOS C SETUP\Ubuntu Server Setup 文件夹中。

(5) 在“浏览 ISO 映像”窗口中,按住鼠标左键拖动左侧列表框的滑块,找到“本地硬盘(D)”,选择“本地硬盘(D)”,然后按住鼠标左键拖动右侧列表框的滑块,找到 HarmonyOS C SETUP 文件夹,然后双击打开该文件夹,如图 3-25 所示。



图 3-23 “安装客户机操作系统”对话框

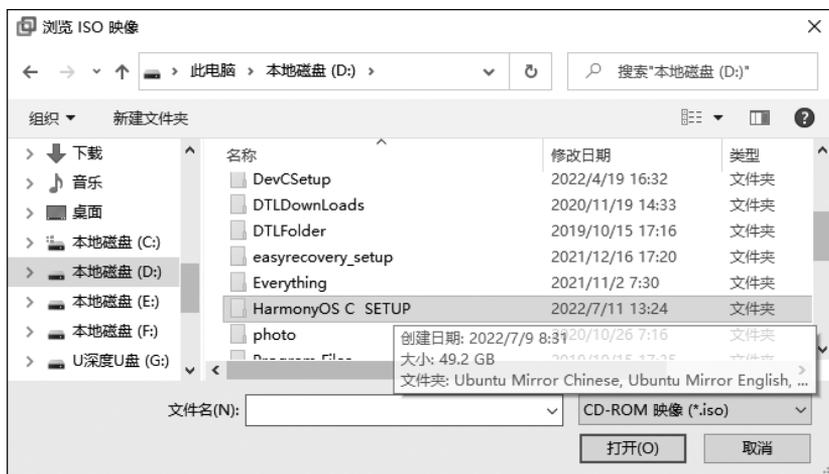


图 3-24 “浏览 ISO 映像”对话框

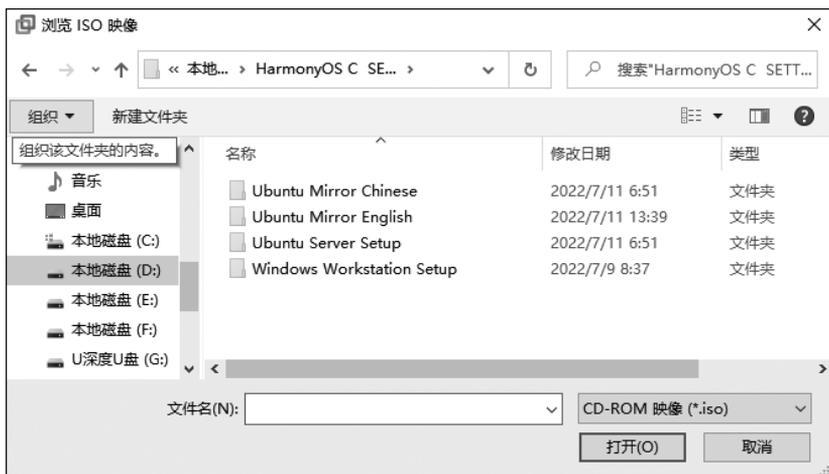


图 3-25 打开 HarmonyOS C SETUP 文件夹

(6) 在如图 3-25 所示的对话框中,双击打开 Ubuntu Server Setup 文件夹,如图 3-26 所示。

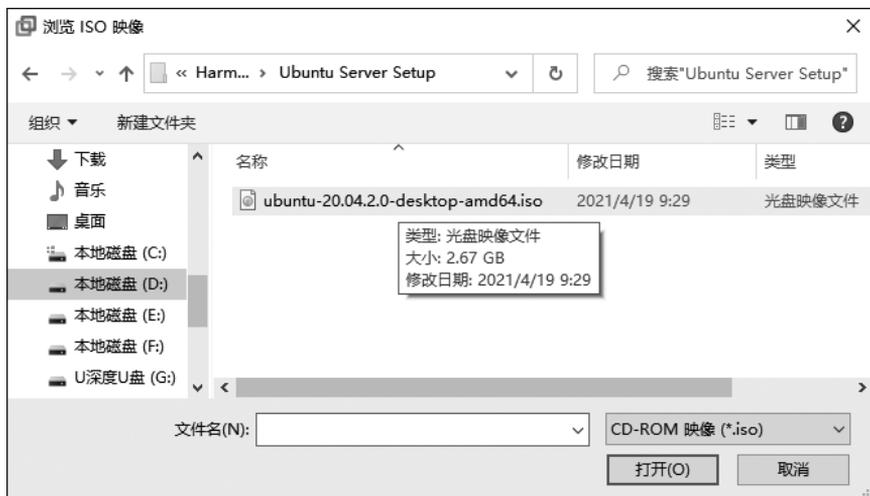


图 3-26 打开 Ubuntu Server Setup 文件夹

(7) 在如图 3-26 所示的对话框中,可以看到已经准备好的 Linux 操作系统 Ubuntu 版本的安装文件 ubuntu-20.04.2.0-desktop-amd64.iso。

注意: 该文件如果保存在其他磁盘的其他文件夹中,可参照此步进行操作,只要找到该文件即可。

(8) 在如图 3-26 所示的对话框中,选择 ubuntu-20.04.2.0-desktop-amd64.iso 文件,然后单击“打开”按钮,返回如图 3-27 所示的“安装客户机操作系统”对话框。然后单击该对话框的“下一步”按钮,出现如图 3-28 所示的“简易安装信息”对话框。



图 3-27 “安装客户机操作系统”对话框

(9) 在如图 3-28 所示的对话框中,在“全名”“用户名”“密码”和“确认”文本框中都输入