

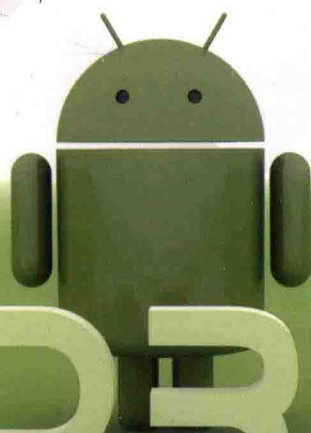
Google Android / App Inventor教材出版资助计划项目

Android

系统软件开发 (底层)

Android XITONG RUANJIAN KAIFA (DICENG)

夏德洲 张 明 主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

Google Android / App Inventor 教材出版资助计划项目

Android 系统软件开发

(底层)

夏德洲 张 明 主 编

朱 波 副主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书是一本介绍 Android 系统核心及系统级应用的技术用书,主要目标是让读者更加深入地理解 Android 系统,让开发过程更高效。目前企业对 Android 人才的需求主要在应用开发和底层系统开发两个方面,企业最紧缺的是兼具两者能力的系统开发工程师。本书共分为 9 章,先介绍 Android 底层的架构和 Android 的启动过程;然后介绍 Android 的系统编译和移植;接下来介绍 Android 的 JNI 和 HAL 层,这是 Android 系统级应用开发的核心内容;最后通过两个实例 Led HAL 和 Sensor HAL 对代码进行详细分析,让读者更加深入地理解 Android 底层系统开发。

本书适合作为高职高专院校嵌入式技术与应用专业、移动互联应用技术专业的教材,也可供中等职业技术学校使用。

图书在版编目(CIP)数据

Android 系统软件开发:底层/夏德洲,张明主编. —北京:
中国铁道出版社,2015.10
ISBN 978-7-113-20718-2

I. ①A… II. ①夏… ②张… III. ①移动终端—应用程序—
程序设计 IV. ①TN929.53

中国版本图书馆 CIP 数据核字(2015)第 226899 号

书 名: Android 系统软件开发(底层)
作 者: 夏德洲 张 明 主编

策 划: 王春霞
责任编辑: 王春霞 鲍 闻
封面设计: 付 巍
封面制作: 白 雪
责任校对: 汤淑梅
责任印制: 李 佳

读者热线: 400-668-0820

出版发行: 中国铁道出版社(100054,北京市西城区右安门西街8号)
网 址: <http://www.51eds.com>
印 刷: 三河市宏盛印务有限公司
版 次: 2015 年 10 月第 1 版 2015 年 10 月第 1 次印刷
开 本: 787 mm×1 092 mm 1/16 印张: 14.25 字数: 342 千
印 数: 1~2 000 册
书 号: ISBN 978-7-113-20718-2
定 价: 32.00 元

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社教材图书营销部联系调换。电话:(010) 63550836

打击盗版举报电话:(010) 51873659

Android 是 Google 公司于 2007 年 11 月发布的一个基于 Linux 内核的开源嵌入式操作系统。经过几年的发展,市场份额迅速壮大,现已跃居全球第一。与此同时,随着行业的发展,Android 研发工程师日益成为 IT 市场的紧缺人才。目前国内的 Android 开发主要以应用开发为主,主要分为两类:(1)企业开发应用;(2)开发通用应用以及游戏开发。第一类开发者一般身处规模较大的公司,这些公司主要为自有品牌或其他品牌设计手机/平板电脑的总体方案。除了根据需求对系统进行定制外,更多的工作在于为这些系统编写定制的应用。第二类开发者,一般处于创业型公司或者是独立开发者。

近几年各大专院校纷纷开设移动互联应用技术专业,但是在专业教学过程中都面临教材难觅、教材内容更新滞后等问题。虽然目前市场上的 Android 开发书籍比较多,但几乎都是针对 Android 应用层的开发。而作为一名合格的 Android 开发工程师,还要了解 Android 的工作机制,这就涉及 Android 的架构;再往下,就是操作系统层级了,这里应该对 Linux 操作系统进行学习,熟悉其内核和运行原理,熟悉 ARM 体系架构及常用指令,并熟悉 Android 的 JNI 和 HAL,掌握其移植方法。针对以上需求我们编写了本教材。

本书从 Android 底层原理开始讲起,结合真实的案例向读者详细介绍 Android 内核、Android 系统移植、Android JNI 调用和 HAL 框架开发流程。全书分为 9 章,依次讲解 Android 源代码的下载、编译,Android 的启动流程、HAL 层深入分析等,重点介绍了与 Android 开发相关的底层知识,并对 Android 源代码进行了剖析。

本书由湖北工业职业技术学院信息与智能工程系夏德洲、张明担任主编并统稿,朱波任副主编。感谢唐攀无私地提供了很多帮助,Google 公司大学合作部的朱爱民经理也对本书提供了技术和资金上的支持,在此表示衷心的感谢。

由于时间仓促,本书可能存在一些不妥之处,请读者见谅并欢迎读者批评指正。

编 者

2015 年 8 月

第 1 章 概述	1
1.1 Android 操作系统介绍	1
1.2 Android 软件架构介绍	2
1.3 Android 子系统介绍	3
1.4 Android 应用程序开发过程	4
小结	5
习题	5
第 2 章 Android 源码开发环境搭建	6
2.1 搭建主机虚拟机环境	6
2.1.1 VMware Workstation 介绍	6
2.1.2 安装 VMware Workstation 虚拟机软件	7
2.1.3 安装 Ubuntu 操作系统	9
2.1.4 VMware 网络配置	15
2.1.5 VMware 与主机数据共享	18
2.1.6 VMware 添加新硬件	20
2.2 搭建 Linux 编译环境	22
2.2.1 建立 Ubuntu 编译环境	23
2.2.2 JDK 安装	23
2.2.3 安装 Android 编译工具	25
2.2.4 下载 Android 源码	25
2.2.5 下载 Linux 内核源码	26
2.3 编译 Android 源码	26
2.3.1 Android 源码目录结构	27
2.3.2 编译 Android	30
2.3.3 编译 Linux 内核	33
2.4 搭建 Android SDK 开发环境	34
2.4.1 下载、安装 Eclipse	34
2.4.2 安装 ADT 插件	34
2.4.3 下载、配置 Android SDK 工具包	37
2.4.4 下载 Android SDK 平台	38
2.4.5 通过 Android SDK Manager 创建模拟器	38
2.4.6 应用程序 Framework 源码级调试	39
2.5 定制 Android 模拟器	45
2.6 实训：Android 4.0 开发环境搭建及源码编译	47

小结	57
习题	57
第 3 章 Android 系统的启动	59
3.1 Android init 进程启动	59
3.2 Android 本地守护进程	65
3.2.1 ueventd 进程	66
3.2.2 adbd 进程	67
3.2.3 servicemanager 进程	67
3.2.4 vold 进程	68
3.2.5 ril-daemon 进程	68
3.2.6 surfaceflinger 进程	68
3.3 zygote 守护进程与 system_server 进程	69
3.3.1 zygote 守护进程的启动	69
3.3.2 zygoteInit 类的功能与 system_server 进程的创建	73
3.3.3 system_server 进程的运行	76
3.3.4 HOME 桌面的启动	82
3.4 实训：通过 Init.rc 脚本开机启动 Android 应用程序	84
小结	86
习题	86
第 4 章 Android 编译系统与定制 Android 平台系统	87
4.1 Android 编译系统	87
4.1.1 Android 编译系统介绍	87
4.1.2 Android.mk 文件	88
4.2 实训：编译 HelloWorld 应用程序	91
4.3 定制 Android 平台系统	93
4.3.1 添加新产品编译项	93
4.3.2 定制产品的意义及定制要点	99
4.4 实训：定制开机界面	100
4.5 实训：定制开机文字	104
4.6 实训：定制系统开机动画	106
小结	109
习题	110
第 5 章 JNI 机制	111
5.1 JNI 概述	111
5.2 JNI 原理	112
5.3 JNI 中的数据传递	114
5.3.1 JNI 基本类型	114
5.3.2 JNI 引用类型	115
5.4 Java 访问本地方法	116

5.5	JNI 访问 Java 成员	117
5.5.1	取得 Java 属性 ID 和方法 ID	118
5.5.2	JNI 类型签名	120
5.5.3	JNI 操作 Java 属性和方法	121
5.5.4	在本地代码中创建 Java 对象	123
5.5.5	Java 数组在本地代码中的处理	124
5.6	局部引用与全局引用	126
5.6.1	局部引用	127
5.6.2	全局引用	128
5.6.3	在 Java 环境中保存 JNI 对象	128
5.7	本地方法的注册	129
5.7.1	JNI_OnLoad 方法	129
5.7.2	RegisterNatives 方法	130
5.8	实训：JNI 调用实训	133
	小结	139
	习题	140
第 6 章	Android 的对象管理	141
6.1	智能指针	141
6.2	轻量级指针	142
6.3	RefBase 类	145
6.4	弱引用指针 wp	150
6.5	智能指针的示例	153
	小结	154
	习题	154
第 7 章	Binder 通信	155
7.1	Android 进程空间与 Binder 机制	155
7.1.1	Android 的 Binder 机制	156
7.1.2	面向对象的 Binder IPC	157
7.2	Binder 框架分析	158
7.2.1	Binder Driver	158
7.2.2	Open Binder Driver	159
7.2.3	ServiceManager 与实名 Binder	161
7.3	Android Binder 协议	162
7.3.1	BINDER_WRITE_READ 之写操作	163
7.3.2	BINDER_WRITE_READ 之从 Binder 读出数据	164
7.3.3	struct binder_transaction_data：收发数据包结构	165
	小结	167
	习题	167

第 8 章	Android HAL 硬件抽象层	168
8.1	Android HAL 介绍	168
8.1.1	HAL 存在的原因	169
8.1.2	Module 架构	169
8.1.3	新的 HAL 架构	170
8.2	HAL Stub 构架	171
8.2.1	HAL Stub 框架分析	171
8.2.2	HAL Stub 注册	172
8.2.3	HAL Stub 操作	174
8.3	Led HAL 实例	177
8.3.1	Led HAL 框架	177
8.3.2	LED HAL 代码架构	179
8.3.3	LED Demo 代码分析	180
8.3.4	LedService 代码分析	181
8.3.5	Led 本地服务代码分析	183
8.3.6	LED HAL 深入理解	188
8.4	实训：基于 Android 4.0 平板的 LED 灯控制	188
	小结	192
	习题	192
第 9 章	HAL 硬件抽象层进阶 Sensor HAL 实例	193
9.1	Android Sensor 架构	193
9.1.1	Android Sensor 框架	193
9.1.2	Android Sensor 工作流程	194
9.2	Sensor HAL 应用程序	198
9.2.1	Sensor HAL 应用程序	198
9.2.2	Android Manager 机制	198
9.2.3	获得 Sensor 系统服务	200
9.3	SensorManager	203
9.3.1	本地 SensorManager 创建	203
9.3.2	获得 SensorService 服务	207
9.3.3	获得 SensorService 监听及事件捕获	210
9.3.4	本地封装类 SensorDevice	213
9.4	Sensor HAL 回顾	217
9.5	实训：SensorDemo 的编译	218
	小结	219
	习题	219

第1章

概述

本章主要介绍 Android 系统的基本特点、系统架构组成及应用开发方式。

学习目标：

- 了解 Android 操作系统。
- 熟悉 Android 软件架构。
- 熟悉 Android 子系统。
- 掌握 Android 应用程序开发过程。

1.1 Android 操作系统介绍

Android 是 Google 公司于 2007 年 11 月发布的一款优秀的智能移动平台操作系统。到 2011 年第一季度，Android 在全球的市场份额首次超过 Nokia 的 Symbian 系统，跃居全球第一。

Android 系统最初由 Andy Rubin 等人于 2003 年 10 月创建。Google 于 2005 年 8 月 17 日收购 Android 并组建 OHA^① 开放手机联盟开发改良 Android，之后逐渐扩展到平板电脑及其他移动平台领域上。



Android 系统是一个基于 Apache License^②，GPL^③ 软件许可的开源手机操作系统，底层由 Linux 操作系统作为内核，我们可以直接从 Android 的官方网站上下载最新的 Android 源码和相关开发工具包。

Android 官方首页：<http://www.android.com/>。

Android 官方开发者首页：<http://developer.android.com/index.html>。

Android 官方开源项目 AOSP 首页：<http://source.android.com/>。

① OHA: Open Handset Alliance 的缩写，是美国 Google 公司于 2007 年 11 月 5 日宣布组建的一个全球性的联盟组织。这一联盟支持 Google 发布的手机操作系统或者应用软件，共同开发名为 Android 的开放源代码的移动系统。开放手机联盟包括手机制造商、手机芯片厂商和移动运营商几类。目前，联盟成员数量已经达到了 34 家，其中包括中国移动、中国联通、中国电信、Haier、Lenovo、华为、中兴通讯等中国软硬件厂商。

② Apache License: 著名的非营利开源组织 Apache 采用的协议。该协议和 BSD 类似，鼓励代码共享和尊重原作者的著作权，允许代码修改、再发布（作为开源或商业软件）。Apache License 是一种对商业应用友好的许可。使用者可以在需要的时候修改代码，并作为开源或商业产品发布或销售。

③ GPL: GNU General Public License 的缩写，是一个广泛被使用的自由软件许可证条款，最初由 Richard Stallman 为 GNU 计划而撰写。GNU 软件许可力图保证用户共享和修改自由软件的自由，即保证自由软件对所有用户是自由的，GPL 给予了计算机程序自由软件的定义，并且使用“Copyleft”来确保程序的自由被完善地保留。

1.2 Android 软件架构介绍

Android 的软件架构采用了分层结构,如图 1-1 所示,由上至下分别为 Application (应用层)、Application Framework (应用框架层)、Android Runtime & Libraries (运行时库和本地库层)、Linux Kernel (内核层),如图 1-1 所示。

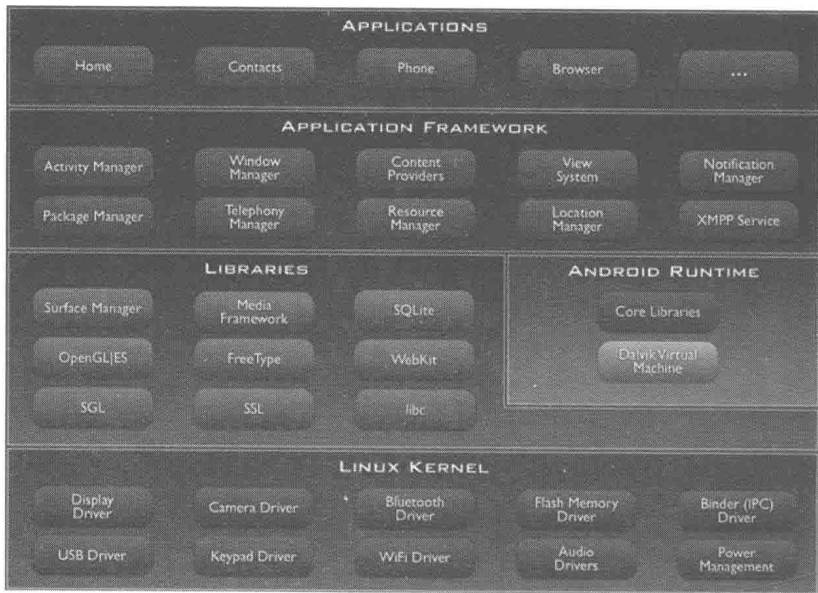


图 1-1 Android 软件架构图

(1) Application (应用层): 用户安装应用程序及系统自带应用程序层,主要用来与用户进行交互,如 Home 指 Android 手机的桌面,Phone 指电话应用,用来拨打电话等。

(2) Application Framework 应用框架层: 系统框架层,封装了大量应用程序所使用的类,从而达到组件重用的目的,它主要向上层应用层提供 API,如 ActivityManager 主要用于管理所有的 Activity 画面导航、回退等与生命周期相关的操作,PackageManager 主要用来管理程序安装包的安装、更新、删除等操作。

(3) Android Runtime & Libraries (运行时库和本地库层): Runtime 是 Android 的运行环境,在该层有 Dalvik Virtual Machine (Android 的虚拟机简称 DVM) 的实现,在 DVM 中运行着 Java 的核心语言库代码和 Java 程序。同时,在 DVM 运行期间要调用系统库代码,如负责显示的 SurfaceManager 本地代码,负责多媒体处理相关的 Media Frameworks 代码及 C 库 libc 等。

(4) Linux Kernel 内核层: Android 系统是基于 Linux 系统的,所以 Android 底层系统相关的框架和标准的 Linux 内核没有什么很大的区别,只不过添加了几个 Android 系统运行必备的驱动,如: Binder IPC 进程间通信驱动、Power Manager 电源管理驱动等。

总结: Android 的软件架构是学习 Android 开发必须要掌握的知识点,它对我们将来编写 Android 应用程序,理解 Android 框架代码,编写本地代码,修改底层驱动都有重要的指导意义,可谓学习 Android 的灵魂。

Android 是一个庞大的手机操作系统，它不仅使手机实现了打电话、发信息等基本功能，还实现了更复杂的多媒体处理、2D 和 3D 游戏处理、信息感知处理等功能。

Android 的子系统如图 1-2 所示。

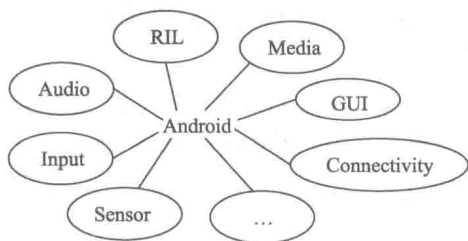


图 1-2 Android 主要子系统

1. Android RIL 子系统

RIL (Radio Interface Layer) 子系统即无线电接口系统用于管理用户的电话、短信、数据通信等相关功能，它是每个移动通信设备必备的系统。

2. Android Input 子系统

Input 子系统用来处理所有来自用户的输入数据，如触摸屏、声音控制物理按键等。

3. Android GUI 子系统

GUI 即图形用户接口，也就是图形界面，它用来负责显示系统图形化界面，使用户与系统及信息交互更加便利。Android 的 GUI 系统和其他各子系统关系密切相关，是 Android 中最重要的子系统之一，如绘制一个 2D 图形、通过 OpenGL 库处理 3D 游戏、通过 SurfaceFlinger^① 来重叠几个图形界面。

4. Android Audio 子系统

Android 的音频处理子系统，主要用于音频方面的数据流传输和控制功能，也负责音频设备的管理。Android 的 Audio 系统和多媒体处理紧密相连，如视频的音频处理和播放、电话通信及录音等。

5. Android Media 子系统

Android 的多媒体子系统，它是 Android 系统中最庞大的子系统，与硬件编解码、OpenCore 多媒体框架、Android 多媒体框架等相关，如音频播放器、视频播放器、Camera 摄像预览等。

6. Android Connectivity 子系统

Android 连接子系统是智能设备的重要组成部分，它除了一般网络连接，如以太网、Wi-Fi^② 外，还包含蓝牙连接、GPS^③ 定位连接、NFC^④ 等。

① Android 中图形混合器，用于将屏幕上显示的多个图形进行混合显示。

② Wi-Fi 全称 Wireless Fidelity，是当今使用最广的一种无线网络传输技术。

③ GPS 是英文 Global Positioning System (全球定位系统) 的简称。

④ NFC 是 Near Field Communication 缩写，即近距离无线通信技术。由飞利浦公司和索尼公司共同开发的 NFC 是一种非接触式识别和互联技术，可以在移动设备、消费类电子产品、PC 和智能控件工具间进行近距离无线通信。

7. Android Sensor 子系统

Android 的传感器子系统为当前智能设备大大提高了交互性, 它在一些创新的应用程序和用户体验中发挥了重要作用, 传感器子系统和手机的硬件设备紧密相关, 如 gyroscope (陀螺仪)、accelerometer (加速度计)、proximity (距离感应器)、magnetic (磁力传感器) 等。

1.4

Android 应用程序开发过程

Android 应用程序开发是基于 Android 架构提供的 API 和类库编写程序, 这些应用程序是完全的 Java 代码程序, 它们构建在 Android 系统提供的 API 之上。

开发 Android 应用程序可以基于 Google 提供的 Android SDK 开发工具包, 也可以直接在 Android 源码中进行编写。

1. Android SDK 开发

它提供给程序员一种最快捷的开发方式, 基于 IDE 开发环境和 SDK 套件, 快速开发出标准的 Android 应用程序, 但是, 对于一些要修改框架代码或基于自定义 API 的高级开发, 这种方式难以胜任。

2. Android 源码开发

基于 Android 提供的源码进行开发, 可以最大限度体现出开源的优势, 让用户自定义个性的 Android 系统, 开发出更高效、更与众不同的应用程序, 这种方式更适合于系统级开发, 对程序员要求比较高, 这也是本书的重点。

Android 源码开发过程:

1. 搭建开发环境

根据两种开发方式的不同, 搭建开发环境略有不同, 本书侧重于系统底层源码开发, 只介绍第二种开发方式。对于第一种方式, 请读者参考相关书籍资料。

2. 下载 Android 源码

得益于 Android 的开源特点, Android 源码中包含大量宝贵的技术知识, 可以在阅读源码过程中更深入地了解 Android 系统的奥秘, 为用户编写更高效、更有特点的应用程序打下基础, 同时能展现给读者一个更庞大系统的设计蓝图, 为系统设计师及项目经理提供参考。同时, Android 的源码中提供的应用程序示例、设计模式、软件架构为用户编写大型应用程序提供经验。

3. 编译 Android 源码

通过编译 Android 源码, 生成开发环境及目标系统, 为用户做系统底层开发、系统定制与优化做准备, 通过分析编译过程, 让用户学习到大型工程的代码管理与编译原理。

4. 配置开发环境安装

为了更有效地进行开发, 通常会对开发环境做配置, 不同的程序员可能会有不同的编程习惯。

小 结

本章主要介绍了 Android 操作系统的基本知识,包括 Android 的架构、Android 的子系统,同时还简述了 Android 应用开发的流程及分类。

通过本章的学习,读者应该能够简单地了解 Android 的架构,了解 Android 的子系统,能够了解 Android 应用程序的开发流程。

习 题

1. Android 软件架构包括哪几层?
2. Android 的子系统主要包括哪些?
3. Android 应用开发分为几种,各自有哪些特点?

第2章

➡ Android 源码开发环境搭建

本章主要讲解如何搭建基于 Ubuntu 的 Android 开发环境，为后面章节的内容讲解做铺垫。

学习目标：

- 掌握安装 VMware 虚拟机软件安装过程。
- 掌握安装 Ubuntu 操作系统安装方法。
- 熟悉 VMware 网络配置。
- 了解 VMware 与主机数据共享。
- 掌握 VMware 添加新硬件方法。
- 掌握 JDK 安装过程。
- 掌握 Android 编译工具使用方法。

2.1 搭建主机虚拟机环境

Android 源码开发可以在 MacOS 上或 Ubuntu 上进行，目前不支持在 Windows 系统下开发，而国内用户主机环境大量使用 Windows 系统，这也意味着用户要在 Windows 系统里安装一个虚拟机软件，将 Ubuntu 虚拟运行在 Windows 系统里。

2.1.1 VMware Workstation 介绍

VMware(中文:威睿)是全球著名的虚拟机软件公司。VMware 工作站(VMware Workstation)是 VMware 公司销售的商业软件产品之一。

VMware Workstation 是一个虚拟 PC 软件，它可以使你在一台机器上同时运行两个或更多 Windows、DOS、Linux 系统。与“多启动”系统相比，VMware 采用了完全不同的概念。多启动系统在一个时刻只能运行一个系统，在系统切换时需要重新启动计算机。VMware 是真正“同时”运行多个操作系统在主系统的平台上，就像标准 Windows 应用程序那样切换。而且每个操作系统都可以进行虚拟的分区、配置而不影响真实硬盘的数据，甚至可以通过网卡将几台虚拟机连接为一个局域网，极其方便。但是，安装在 VMware 中的操作系统在性能上比直接安装在硬盘上的物理主机系统低不少，所以通常用于程序开发或测试。

运行 VMware Workstation 的计算机和操作系统被称为宿主机(host)。在一个虚拟机中运行的操作系统实例被称为虚拟机客户(guest)。

由于与宿主机的真实硬件无关，所有虚拟机客户使用相同的硬件驱动程序，虚拟机实例是对各种计算机高度可移植的。例如，一个运行中的虚拟机可以被暂停下来，并被复制到另外一台作为宿主的真实计算机上，然后从其被暂停的确切位置恢复运行。

VMware Workstation 特点：

- 客户机以程序的方式运行在宿主机中。
- 宿主机与虚拟机客户机完全并列平等运行，没有从属关系。
- 宿主机与虚拟机可以通过配置方便地实现文件共享、网络连接、硬件配置等操作。
- 客户机的硬件和宿主机环境无关，都是通过软件仿真的。
- 客户机具有可移植性，可以在不同宿主机上运行。
- 客户机性能比物理主机低得多。

2.1.2 安装 VMware Workstation 虚拟机软件

VMware Workstation 目前常见版本是 v8.0.x，推荐使用较新版本 v8.0.x 以上版本，以保证 Ubuntu 的正常使用。

(1) 双击安装包（见图 2-1）进行安装。

(2) 安装包启动界面如图 2-2 所示。

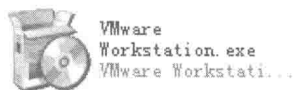


图 2-1 安装 VMware 安装包

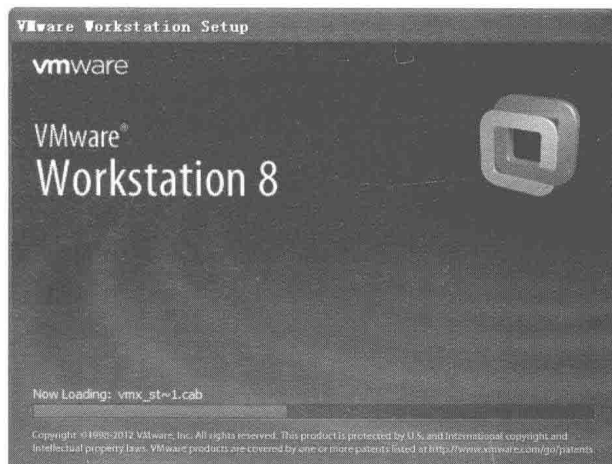


图 2-2 安装界面启动

(3) 单击 Next 按钮开始安装，如图 2-3 所示。

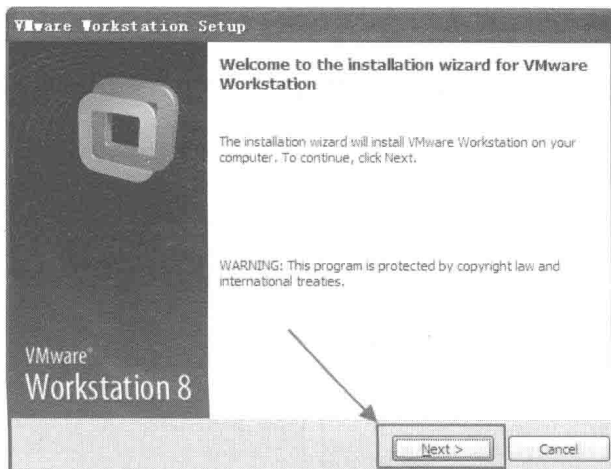


图 2-3 安装界面

(4) 单击 Typical 按钮, 选择典型安装方式, 使用默认配置, 如图 2-4 所示。

(5) 单击 Next 按钮, 设置安装目录, 如图 2-5 所示。

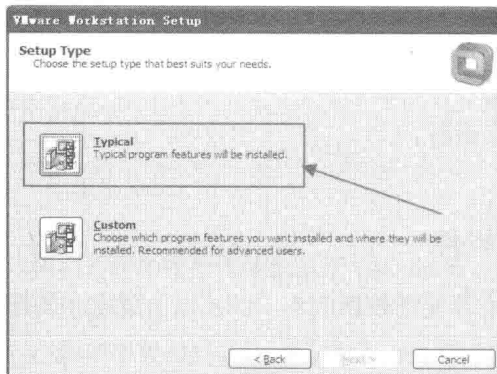


图 2-4 选择“典型”安装

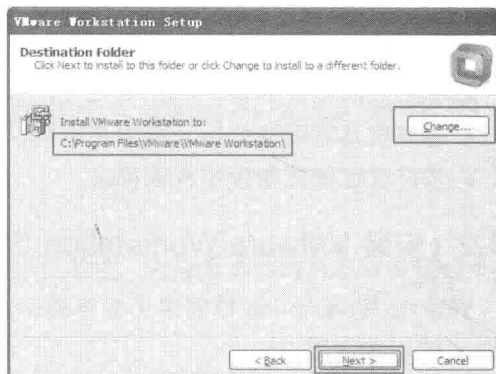


图 2-5 选择安装目录

(6) 单击 Next 按钮, 使用默认选项, 如图 2-6 所示。

(7) 单击 Next 按钮开始安装, 如图 2-7 所示。

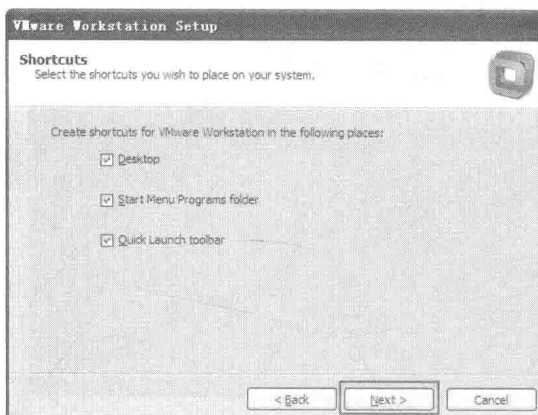


图 2-6 选择启动菜单界面

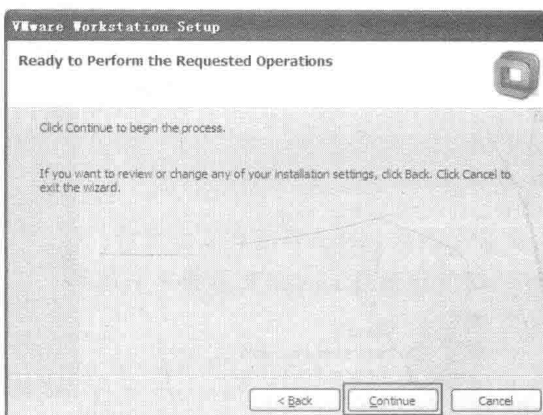


图 2-7 安装界面

(8) 安装完成, 如图 2-8 所示。

(9) 选择 Yes 单选按钮, 接受 VMware 的许可, 如图 2-9 所示。

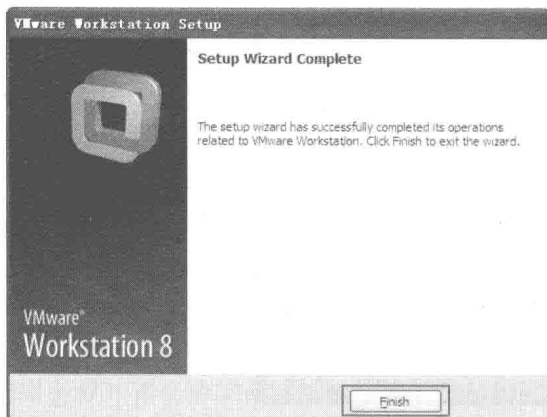


图 2-8 安装完成界面

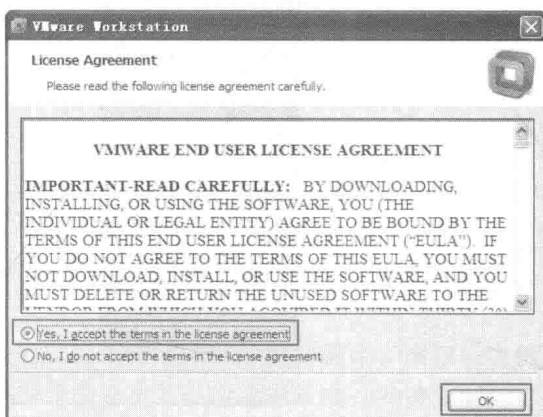


图 2-9 接受安装许可

(10) 安装完毕后启动 VMware Workstation 8, 如图 2-10 所示。

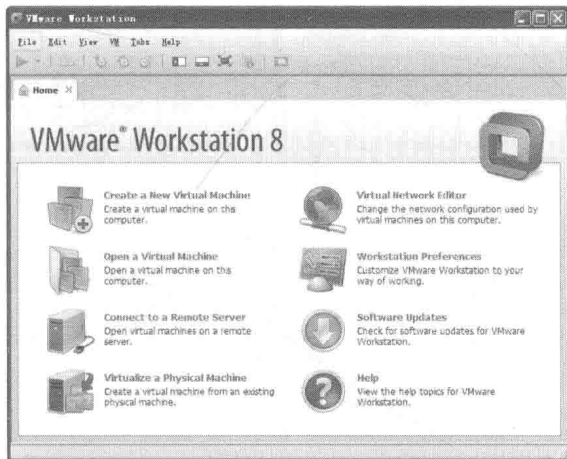


图 2-10 VMware Workstation 首界面

2.1.3 安装 Ubuntu 操作系统

安装完 VMware Workstation 之后, 就可以安装虚拟机了。

(1) 首先从 Ubuntu 的官方网站下载 Ubuntu LTS^①版本 ISO 映像文件。

Ubuntu 中文官方网站: <http://www.ubuntu.org.cn/>。

Ubuntu 桌面版下载地址: <http://www.ubuntu.com/download/desktop>。

Ubuntu 操作系统简介:

Ubuntu 基于 Debian 发行版和 GNOME 桌面环境, 与 Debian 的不同在于它每 6 个月会发布一个新版本。Ubuntu 的目标在于为一般用户提供一个最新的、同时又相当稳定的主要由自由软件构建而成的操作系统。Ubuntu 具有庞大的社区力量, 用户可以方便地从社区获得帮助。

(2) 创建一个新的虚拟机, 如图 2-11 所示。

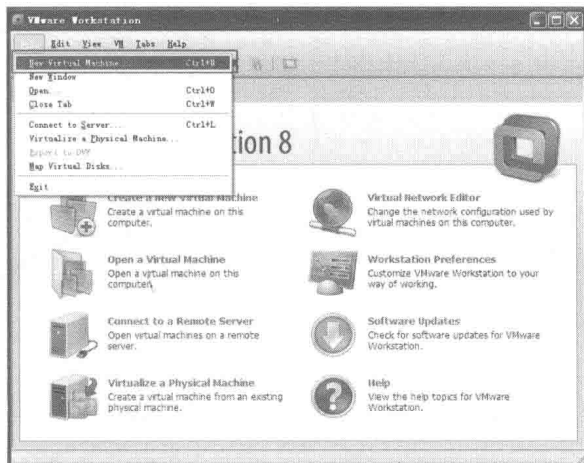


图 2-11 创建新虚拟机

^① Ubuntu 操作系统分为一般发行版和长期支持 (Long Term Support, LTS) 版, Ubuntu 将为每个版本提供至少 18 个月的支持并致力于在该发行版的支持期内持续发布安全和关键补丁。而社区会为 LTS 版提供至少三年的技术支持, 因此, 强烈建议用户下载使用 LTS 版以得到 Ubuntu 社区的更新和支持。