嵌入式Linux开发详解

——基于AT91RM9200和Linux 2.6

刘庆敏　张小亮　编著

北京航空航天大学出版社

嵌入式 Linux 开发详解

——基于 AT91RM9200 和 Linux 2.6

刘庆敏　张小亮　编著

北京航空航天大学出版社

内 容 简 介

本书介绍了嵌入式 Linux 开发需要掌握的基础知识，采用分层的方法对关键技术进行了详细的讲解，且辅以大量实例。共分为 7 章。第 1、2 章介绍嵌入式系统和 Linux 的基础知识。第 3～7 章从实践的角度分层次介绍嵌入式 Linux 开发的流程和关键技术。其中，第 3 章介绍硬件平台；第 4 章介绍 Boot Loader 的基础理论，对 U-boot 的移植、代码分析、关键技术情景分析等进行了深入探讨；第 5 章介绍了 Linux 内核移植需要具备的知识，重点分析了内核映像格式以及 Boot Loader 与内核的通信机制；第 6 章在介绍嵌入式文件系统的基础上，设计并实现了一个嵌入式混合文件系统；第 7 章介绍了嵌入式开发环境的搭建，并简单介绍了一个数据网关的实例。

本书内容可操作性强，适合嵌入式 Linux 开发初学者参考，也可以作为高等院校有关嵌入式系统开发与应用的实验参考书。

图书在版编目(CIP)数据

嵌入式 Linux 开发详解：基于 AT91RM9200 和 Linux 2.6/刘庆敏等编著. --北京：北京航空航天大学出版社，2010.5

ISBN 978-7-5124-0071-9

Ⅰ.①嵌… Ⅱ.①刘… Ⅲ.①Linux 操作系统—程序设计 Ⅳ.①TP316.89

中国版本图书馆 CIP 数据核字(2010)第 070910 号

嵌入式 Linux 开发详解

——基于 AT91RM9200 和 Linux 2.6

刘庆敏　张小亮　编著

责任编辑　董立娟

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191)　http://www.buaapress.com.cn

发行部电话：010-82317024　传真：010-82328026

读者信箱：bhpress@263.net　邮购电话：(010)82316936

北京市媛明印刷厂印装　各地书店经销

*

开本：787×960　1/16　印张：16　字数：358 千字

2010 年 5 月第 1 版　2010 年 5 月第 1 次印刷　印数：4 000 册

ISBN 978-7-5124-0071-9　定价：29.00 元

前言

曾经梦想成为一名作家，就像喜欢的余秋雨、路遥、霍达一样，思想跃动于笔端。不过似乎总是缺少那么一份灵性，文学之路与我渐行渐远。幸运的是，缘于技术，拜 ChinaUnix 所赐，有了这本书，也因此圆了一个儿时的梦想。

记得是从 2006 年夏天开始接触嵌入式系统，学习 ARM、Linux，虽然忙碌但很充实。在学习的间隙，本着“好记性不如烂笔头”的原则，想要把所学都记录下来。但是传统的纸笔记录太慢，有时候难以把问题的场景清晰而又完整地记录下来，就寻找合适的网络记录手段，于是就有了笔者的博客。

开始纯粹是自己的总结笔记，没想到网友的评价还不错，博客的浏览量提高了，也因此交到了很多朋友。

在本书的编写过程中，得到了很多人的支持和帮助。

首先感谢与非网的 Demi，编辑了笔者写的有关 AT91RM9200 开发的技术专题，放在与非网的图书专栏里，网址如下：http://www.eefocus.com/html/09-04/985330120437N7Rd.shtml。虽然在格式上尚存不足，内容上也比较浅显，但都是自己思考的心血。

还要感谢北京航空航天大学出版社。有了他们的支持，才能有了这本书的问世。

感谢合作伙伴张小亮，他以丰富的项目经验弥补了我的不足。

感谢陆小珊及田岚老师，给我提供了学习的环境，并且教给我好多做人做事的道理。

感谢济南雷森科技有限公司的张宝利、王军、邵宏强，是他们把我带入嵌入式系统的大门，在技术上毫无保留，让我快速成长。

感谢傅炜（网名 Tekkaman Ninja，技术博客 http://tekkman.cublog.cn），不仅对本书提出了好多建议，而且贡献了 5.3 和 5.5 两节。通过与其进行技术交流，修正了原先理解的误区，使得本书更为严谨。

感谢陈琦，给予我诸多的建议，在我写书烦躁之时陪我聊天、鼓励我，让我能够顺利地完成

本书的编写。

最后感谢我的家人以及所有关心我的人！

因为本人水平有限，编写书稿时间很短，书中难免有不当之处，敬请广大读者批评指正。有兴趣的读者可以到我的博客 http://piaoxiang.cublog.cn 上讨论问题，进行技术交流，希望能够共同提高，共同进步。

作 者

2010 年 1 月

目 录

第1章

嵌入式系统设计概述

本章目标

- 了解嵌入式系统的概念、发展历史和特点；
- 了解嵌入式设计的方法；
- 掌握学习嵌入式系统的方法。

嵌入式系统设计是一个极为宽泛的领域，实践性很强。但是若因此抛开基础理论，终究是事倍功半。本书提倡的原则是理论与实践相结合，建议如下：

- 跳过本章，从第 2 章开始，根据书中步骤进行操作，在实践中思考嵌入式系统的概念、组成、开发模型等理论知识，形成自己的认识，然后对照本章，以求把知识点理论化、系统化。
- 走马观花浏览本章，建立嵌入式系统设计的知识体系树。在后续章节的学习中，以此为主线，不断丰富完善知识体系树。

这两种方法都是可行的，对初学者尤其适用。在开始嵌入式学习前，笔者提醒初学者，凡事都不可能一蹴而就，要循序渐进，所以不要心急，静下心来，打好基础，逐步提高，努力终会有所回报。

1.1 嵌入式系统的定义

嵌入式系统是当今最为热门的研究领域之一，它的发展势头已经引起社会各界的广泛关注。从市场的观点看，PC 已经从高速增长过渡到平稳发展时期，其年增长率由 20 世纪 90 年代中期的 35%逐年下降，使单纯由 PC 带领电子产业蒸蒸日上的时代成为历史。根据 PC 时代的概念，美国 Business Week 杂志提出了“后 PC 时代”的概念，即计算机、通信和消费产品的技术结合起来，以 3C 产品的形式通过 Internet 进入家庭，这将是一个极为庞大的嵌入式应用市场。在当今社会中，嵌入式系统已经广泛渗透到了人们工作、生活中的各个领域。

那么什么是嵌入式系统呢？现在业界还没有明确统一的定义。这里只提供几种大家公认的定义。

根据 IEEE(国际电气和电子工程师协会)的定义，嵌入式系统是“控制、监视或者辅助设

备、机器和车间运行的装置”。这主要是从嵌入式系统的用途方面进行定义的。

目前国内一个普遍被认同的定义是：以应用为中心，以计算机技术为基础，软硬件可裁减，适应应用系统对功能、可靠性、成本、体积、功耗严格要求的专用计算机系统。它具备“嵌入性”、“专用性”与“计算机系统”3个基本要素。从这个定义可以看出，我们日常生活中常用的手机、PDA、MP3/MP4、机顶盒等都属于嵌入式系统设备，嵌入式系统已经进入了人们生活的各个方面。

下面从发展历史、组成和特点3个方面，对嵌入式系统这个基本概念进行简单的论述。

1.1.1 嵌入式系统的发展历史

嵌入式系统并非新生事物，已经经过30年的发展历程，只是在发展初期，还没有提出嵌入式系统的概念而已。嵌入式系统的发展主要经历了4个阶段：

(1) 无操作系统阶段

嵌入式系统的最初阶段是基于单片机，具有监测、伺服和设备指示等功能，通常应用于专用性强的工业控制系统；一般没有操作系统的支持，通过汇编语言对系统进行直接控制。这些装置初步具备了嵌入式的应用特点，但是仅仅使用8位CPU芯片来执行一些单线程的程序，因此严格上来说还谈不上“系统”的概念。这一阶段嵌入式系统的主要特点是：系统结构和功能相对单一，处理效率较低，存储容量较小，几乎没有用户接口。由于这种嵌入式系统使用简单、价格低，因此普遍应用于国内工业控制领域，但是现在无法满足现代工业控制和新兴信息家电等领域的需求。

可见，嵌入式系统并非庞然大物，我们平常使用的51单片机正是嵌入式系统的最初阶段。有了单片机的基础，要提高也并非难事。

(2) 简单操作系统阶段

这一阶段的嵌入式系统以嵌入式CPU为基础、以简单操作系统为核心，主要特点是：出现了大量具有高可靠性、低功耗的嵌入式CPU(如PowerPC等)，但是通用性比较弱；操作系统达到一定的兼容性和扩展性，内核精巧且效率高；应用软件较专业化，用户界面不够友好。

(3) 嵌入式操作系统阶段

20世纪90年代，随着硬件实时性要求的提高，嵌入式系统的软件规模不断扩大，逐渐成为主流。主要特点是：嵌入式操作系统能够运行于各种不同类型的微处理器/微控制器上，兼容性好；操作系统内核小、效率高，并且具有高度的模块化和扩展性；具备文件和目录管理、设备管理、多任务、网络、图形用户界面等功能，并且提供了大量的应用程序接口，开发应用程序较简单。

(4) 面向Internet阶段

这是一个正在迅速发展的阶段。目前，大多数嵌入式系统还孤立于Internet之外。21世纪是网络时代，随着Internet的进一步发展以及Internet技术与信息家电、工业控制技术等的

结合日益紧密，嵌入式设备与 Internet 的结合将代表嵌入式系统的未来。

1.1.2 嵌入式系统的组成

嵌入式系统有两大核心技术：硬件核心和软件核心。其中，硬件核心为嵌入式微处理器(Embedded Micro-Processor Unit，EMPU)或者嵌入式微控制器(Embedded Micro-Controller Unit，EMCU)，软件核心为嵌入式操作系统(Embedded Operating System，EOS)。比如本书基于 ARM＋Linux，其实也是根据这两个核心来制定的。这里，EMCU 为 ARM，EOS 为 Linux。所以建议大家在开始就建立起这个概念，围绕着这两个核心去展开学习。

下面从分层模型的角度进行介绍。

嵌入式系统的组成，如图 1.1 所示。从最通用的角度来看，可以分为 3 个基本层面：硬件层、系统软件层、应用软件层。在硬件层和系统软件层之间，有 Boot Loader 层。从这 4 个层面去理解嵌入式系统是最自然的方式。

(1) 硬件层

硬件层包括嵌入式微控制器和外围设备。其中，嵌入式微控制器是嵌入式系统的核心部分。它与通用处理器最大的区别在于，嵌入式微控制器大多工作在为特定用户群所专门设计的系统中，它将通用处理器中许多由板卡完成的任务集成到芯片内部，从而有利于嵌入式系统在设计时趋于小型化，同时还具有很高的效率和可靠性。如今，嵌入式微控制器的种类繁多，流行的体系结构有 30 多个系列，其中以 ARM、PowerPC、MC 68K、MIPS 等使用最为广泛。

外围设备是嵌入式系统中用于存储、通信、调试、显示等辅助功能的其他部件。目前，常用的嵌入式外围设备按照功能可以分为存储设备(分为易失性存储介质和非易失性存储介质两类)、通信设备(如 RS232、SPI、I^2C、Ethernet 接口等)和显示设备(如 LCD 等)3 类。

固件是适应技术发展而出现的一种手段，既不是硬件，也不是软件。它形态上是硬件，但是功能上是软件，具有两个特点：非易失性载体和固化特性。很多时候，感觉不到 FirmWare 的存在，一般把固件归结为硬件层，比如 AT91RM9200 内部本身有 128 KB 的片内 ROM，固化了一个 Boot Loader 和 Up-Loader，用来支持程序的下载和引导，这部分就属于固件。

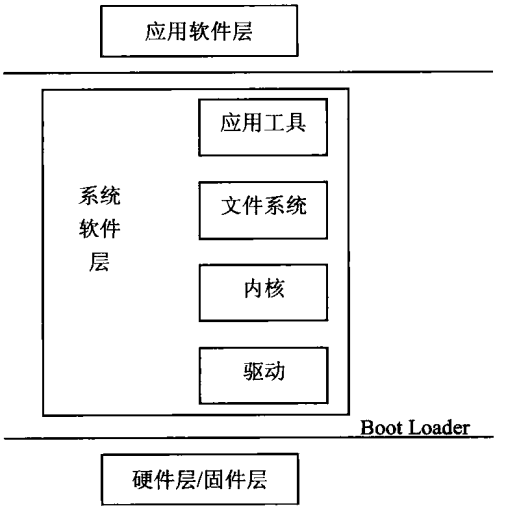

图 1.1 嵌入式系统的组成

(2) Boot Loader 层

该层也称为中间层，是操作系统内核运行前的一段程序，用于完成硬件设备的初始化，加载内核，

为最终调用系统内核做好准备。常见的 Boot Loader 有 U-boot、Blob、Redboot 等。一旦内核加载完成，它的使命也就完成了。

也有一种观点认为，把 Boot Loader 及操作系统的驱动部分看作一体，称为硬件抽象层(Hardware Abstract Layer，HAL)或者板级支持包(Board Support Package，BSP)。这样就将系统上层软件与底层硬件分离开来，使得系统的底层驱动程序与硬件无关，上层软件开发人员无需关心底层硬件的具体情况，根据 BSP 层提供的接口即可进行开发。

(3) 系统软件层

系统软件层由嵌入式操作系统、文件系统、图形用户接口及通用组件模块组成。

嵌入式操作系统从嵌入式发展的第三阶段起开始引入，是嵌入式应用软件开发的基础和开发平台。几种主流嵌入式操作系统有 Windows CE、Linux、VxWorks、QNX、Palm OS 等，都有自己的市场。学习时往往选择一个方面去精深，“专心做好一件事”。比如你可以选择 ARM+Linux，也可以选择 ARM+VxWorks 或者 ARM+Windows CE。

嵌入式文件系统比较简单，主要提供文件存储、检索和更新等功能，一般不提供保护和加密等安全机制。其往往具备以下特点：

- 兼容性。嵌入式文件系统通常支持几种标准的文件系统，如 FAT32、JFFS2、CramFS、YAFFS 等。
- 实时文件系统。除支持标准的文件系统外，为提高实时性，有些嵌入式文件系统还支持自定义的实时文件系统，这些文件系统一般采用连续的方式存储文件。
- 可裁减、可配置。根据嵌入式系统的要求选择所需的文件系统及存储介质，配置可同时打开的最大文件数等。
- 支持多种存储设备。嵌入式系统的外存形式多样，嵌入式文件系统需要方便地挂接不同存储设备的驱动程序，具有灵活的设备管理能力。同时，根据不同外部存储器的特点，嵌入式文件系统还需要考虑其性能、寿命等因素，发挥不同外存的优势，提高存储设备的可靠性和使用寿命。

(4) 应用软件层

应用软件层与应用场合有关。针对复杂的系统，在系统设计初期就要对系统进行需求分析，确定系统的功能，然后设计选择相应的应用组件，以求达到最合理配置。这是最能体现产品差异化的层次。

这 4 个层次并非存在绝对的界限，读者应在实践中体会。

1.1.3 嵌入式系统的特点

嵌入式系统有自己独特的地方，这里可以列举几点，以加深对嵌入式系统的认识。需要注意的是，不能通过某几个特点来判定是否为嵌入式系统。推荐毛德操和胡希明合著的经典《嵌入式系统——采用公开源代码和 StrongARM/XScale 处理器》，其中对嵌入式系统的特点论

述非常精辟。

① 嵌入式系统的软件代码尤其要求高质量、高可靠性。因为嵌入式系统常常在无人照看的条件下运行，有的甚至是在人迹罕至的地方运行，因此，其代码必须有更高的要求，而且通常采用一种称为“看门狗”(WatchDog)的机制，这也是一般通用计算机中没有的。

② 面向特定应用的特点。嵌入式系统与通用型系统的最大区别就在于嵌入式系统大多工作在为特定用户群设计的系统中，因此它通常具有功耗低、体积小、集成度高等特点，并且可以满足不同应用的特定需求。

③ 嵌入式系统一般都不带用于大容量存储的外部设备，也就是不带磁盘。而操作系统的映像和可执行程序一般都存放在非易失性存储介质中，比如 ROM 或者 Flash；不过有些应用于通信领域的高端设备不符合该特点。

④ 许多嵌入式系统的人机界面也有特殊性。许多嵌入式系统都不提供图形人机界面，而只是提供一个面向字符的控制台接口。不过往往还带有如小型的 LCD 显示屏、发光二极管等的辅助显示设备，甚至报警装置。在工业控制领域，大多是简单的面向字符的控制台接口，而且仅仅是开发人员调试监控的时候才会使用。对终端用户而言，他们是不关心的。

⑤ 嵌入式系统本身不具备二次开发能力，即设计完成后用户通常不能对其中的程序功能进行修改，必须有一套开发工具和环境才能再次开发。

关于嵌入式系统的特点，读者必须结合实践经验，形成自己的认识。笔者虽然列举了几条特点，但是希望大家先否定，验证后再考虑是否肯定。带着疑问去学习，由点及面，才能举一反三，进步才能更迅速。

1.2 嵌入式系统设计概述

嵌入式系统设计的主要任务是定义系统的功能，决定系统的架构，并将功能映射到架构。这里的架构既包括硬件系统架构，也包括软件系统架构。嵌入式系统的设计方法不同于一般的硬件设计、软件设计，而是采用软/硬件协同的方法。开发过程会涉及软件、硬件及其相关应用的专业知识。和通常的系统设计相比，嵌入式系统的设计具有以下几个特点：

(1) 软/硬件协同开发

软/硬件协同开发就是在整个设计的生命周期，软件和硬件的设计一直是保持并行的，在设计过程中两者交织在一起，互相支持，互相提供开发的平台，而不是传统方法中将软/硬件设计分开独立进行。在设计流程的开始就要将系统所要实现的功能划分到硬件或软件实现，然后独立进行硬件和软件设计，最后才进行软/硬件的集成。

(2) 需要交叉编译

简单的说，交叉编译就是在一个平台上生成在另一个平台上执行的代码。这里的平台包括体系结构(Architecture)和操作系统(OS)。同一个体系结构可以运行不同的操作系统，同

样，同一个操作系统也可以在不同的体系结构上运行。举例来说，x86 Linux 平台是 Intel x86 体系结构和 Linux for x86 操作系统的统称。

为什么要采用交叉编译呢？原因有两个。一是目标平台所需要的 Boot Loader 以及 OS 核心还没有建立时，需要做交叉编译。二是目标机设备不具备一定的处理器能力和存储空间，即单独在目标板上无法完成程序开发，所以只好求助宿主机。这样可以在宿主机上对即将在目标机上运行的应用程序进行编译，生成可以在目标机上运行的代码格式，然后移植到目标板上，也就是目前嵌入式程序开发的 Host/Target 模式。

(3) 嵌入式系统的程序需要固化

通用的系统在测试完成后就可以直接投入使用，其目标环境一般是 PC 机，因此在总体结构上与开发环境差别不大。而嵌入式系统的开发环境是 PC 机，但是应用软件在目标环境下必须存储在非易失性存储器中，保证用户关机后下次能够再次使用。因此，在系统应用软件开发完成之后，应生成固化版本。

此外，嵌入式系统还需要提供强大的硬件开发工具和软件包的支持，需要设计者从速度、功能和成本综合考虑。此外，嵌入式系统对稳定性、可靠性、功耗、抗干扰性等方面的性能要求都比通用系统的要求更为严格，所以相对而言，嵌入式系统的软件开发难度更大一些。

1.3 嵌入式系统的学习方法

嵌入式系统是一门交叉学科，入门的门槛比较高。在学习过程中要注意一定的方法，才能够起到事半功倍的作用。常用的方法有：抓住本质，先主后次；分层整合；协同分工；情景分析。例如，根据分层整合的方法建立知识体系树，随着项目实践和学习的深入，不断地去丰富知识体系树。在分层与整合中，达到学习的最优化。大家可以参考 http://piaoxiang.cublog.cn，结合自己的情况建立知识体系树。

在后面的章节中，笔者会以实践为主，根据嵌入式系统的开发流程展开详述。

本章总结

本章从嵌入式系统的定义入手，介绍了嵌入式系统的发展历史、组成和特点。结合实际开发经验，总结了比较实用的学习方法，给出了知识体系树。

从理论的角度讲，本章只是简单地给出了几个概念，并没有深入探讨。希望大家在后续章节的学习中，能够根据自己的实践去形成自己的认识，以达到比较理想的效果。

第2章

磨刀不误砍柴工

本章目标

- 了解 Linux 是什么；
- 学会如何安装 Linux；
- 学会如何设置 Red Hat Linux 9；
- 掌握 shell 的基本用法，提高效率；
- 掌握 vim、gcc、gdb、make 等开发工具；
- 掌握嵌入式环境中常用的命令。

“磨刀不误砍柴工”，所以在进入 ARM＋Linux 的嵌入式世界探宝之前，最好先打好 Linux 的基础。例如，了解 Linux 的常识，掌握安装方法，会常用的命令，能够完成简单程序的编辑、编译、执行与调试。在内核编译阶段，对 Linux 的要求会更高。只有掌握了基础，才能更有信心去迎接挑战。

2.1 Linux 概述

用一句话来概括，Linux 是一套类似于 Unix 的操作系统，包括内核、系统工具、应用程序以及完整的开发环境。严格来说，Linux 只表示 Linux 内核，不过大家都已经习惯了用 Linux 来形容整个基于 Linux 内核并且使用 GNU 工程各种工具的操作系统。

Linux 是一个易于移植程序的出色平台，在嵌入式系统领域得到了广泛应用。详细请参考 http://zh.wikipedia.org/wiki/Linux。

提示

Linux 的内核版本号：

到 2009 年 7 月 30 日，Linux 内核的最新版本是 2.6.30.4。

Linux 内核版本号格式为 major.minor.patch，major 代表主版本号，minor 代表次版本号，如为奇数，代表测试版本，为偶数，代表稳定版本；patch 代表扩展版本号。其中，major 和

minor 标志着重要的功能变动，patch 代表较小的功能改动。有些会出现第 4 个数字，这代表更小的 bug 修复，一般是在 major. minor. patch 的基础上提供一个补丁文件。可以从 http://www. kernel. org/下载最新的内核代码。

在标准内核的基础上，有些公司会对其进行扩展，以适应某种特定的应用，这时就采用 major. minor. patch-www 的形式。比如 Russell King 维护的 ARM Linux 版本为 major. minor. patch-rmk。在 ARM Linux 开发中，可从 http://www. kernel. org 下载标准内核(假定版本为 2. 6. 20)，然后从 http://www. arm. linux. org. uk 下载对应内核版本的-rmk 补丁(2. 6. 20-rmk5)。

GNU：

GNU 是 GNU's Not Unix 的缩写，由 Richard Stallman 在 1983 年 9 月 27 日发起，目标是创建一套完全自由的操作系统。GNU 计划开发了大量的自由软件，比如文字编辑器 Emacs、C 语言编译器 GCC 等。为保证其能够自由地“使用、复制、修改和发布”，所有 GNU 软件都必须遵循 GPL 协议(GNU General Public License，GNU 通用公共许可证)。1991 年 Linus Torvalds 编写出 Linux 内核并在 GPL 协议下发布，至 1992 年 Linux 与其他 GNU 软件结合，完全自由的操作系统正式诞生。所以该系统往往称为 GNU/Linux 或者简称 Linux。

2.2 Linux 的安装

实际编写程序是学习一门新语言的好方法。同样，学习 Linux 比较可行的方法就是使用它。首先必须要进行 Linux 的安装。虽然安装 Linux 是一件容易的事情，不过作为初学者，对 Linux 还不熟悉，犯错是难免的，所以不建议直接在电脑上安装 Linux。这里推荐 VMware，因为它不会影响操作系统的运行。

提示

虚拟机：

虚拟机是一种严密隔离的软件容器，可以运行自己的操作系统和应用程序，就好像一台物理计算机。虚拟机的运行完全类似于一台物理计算机，它包含自己的虚拟(即基于软件实现的)CPU、RAM 硬盘和网络接口卡。

操作系统无法分辨虚拟机与物理机之间的差异，应用程序和网络中其他计算机也无法分辨。即使是虚拟机本身也认为自己是一台“真正的”计算机。不过，虚拟机完全由软件组成，不含任何硬件组件。因此，虚拟机具备物理硬件所没有的很多独特优势。

更多关于虚拟化和虚拟机的知识可以参考 http://www. vmware. com/cn/。

Linux的发行版本众多，目前已经超过250个，比较知名的有Red Hat、Debian、Ubuntu、Mandrake等。根据个人爱好不同，选择的发行版也不尽相同。对初学者而言，优先选择使用人群最多的版本。这样遇到问题时，可以方便地找到答案。Red Hat Linux 9在国内的应用非常广泛，而且Red Hat公司的Docs库提供了完备的资料，因此，本书以Red Hat Linux 9作为安装实例来进行讲解。相关的文档，读者可以从Red Hat的Docs库中下载：http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/。

2.2.1 创建一个新的虚拟机

VMware的启动界面如图2.1所示。

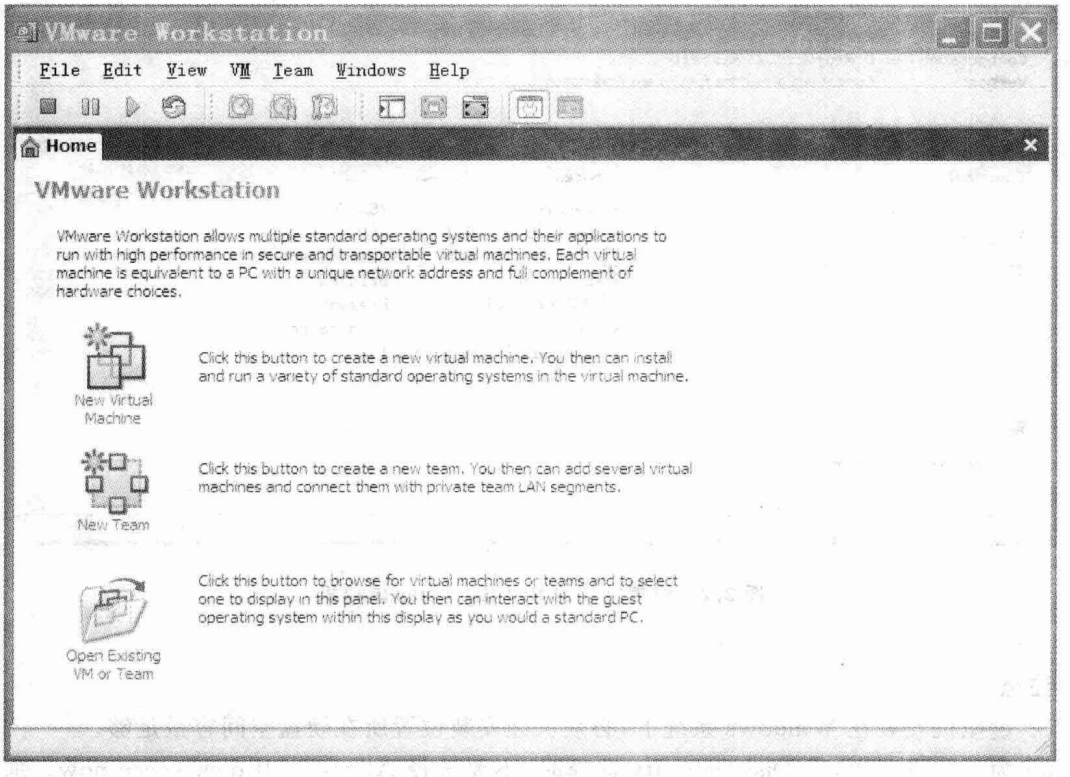

图2.1 VMware启动界面

单击New Virtual Machine，按照提示来创建一个新的虚拟机。步骤如下：

① 单击“下一步”，在Virtual machine configuration中选择Typical。

② 单击“下一步”，在Guest operating system中选择Linux，对应的版本选择Red Hat Linux。

③ 单击“下一步”，选择虚拟机的名字和安装的位置。