

J A V A 系 列 丛 书



J
A
V
A
嵌
入
技
术

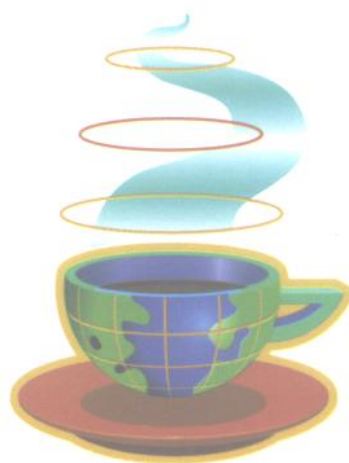
清
华
大

TP312
7260-8

JAVA

嵌入技术

王克宏 主编
徐剑军 徐 鹏 编著



清 华 大 学 出 版 社

<http://www.tup.tsinghua.edu.cn>

T1-1

420-8

Java 系列丛书之十

Java 嵌入技术

王克宏 主编
徐剑军 徐 鹏 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

在当今的信息化浪潮中,不仅有计算网络化的趋势,还有着计算嵌入化的趋势。Java 源于嵌入计算,成长于网络计算。在这两个趋势关系日益密切的情况下,Java 的地位日益加强。要全面了解 Java,就必须了解 Java 嵌入技术。

本书较详细地介绍了 Personal Java、Java Card 平台和相应的编程。读者还可以详细地了解 Java 芯片的结构、特点和应用。需要强调的是,本书目的不仅仅在于介绍技术本身,为了使读者对 Java 嵌入技术在整个信息化浪潮中的广阔的应用前景和不可代替的地位有一个较深刻的理解,本书对 Java 嵌入技术对整个信息化浪潮的影响作了分析,并介绍了 Java 嵌入技术的广阔的应用前景。

本书的读者有一个较广的范围。包括大专院校师生及信息产业界的技术人员。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无防伪标签者不得销售。

书 名: Java 嵌入技术

作 者: 王克宏主编 徐剑军 徐 鹏编著

出 版 者: 清华大学出版社(北京清华大学校内,邮政编码:100084)

责任编辑: 徐培忠

印 刷 者: 北京市清华园胶印厂

发 行 者: 新华书店总店北京科技发行所

开 本: 787×1092 1/16 印张: 7.5 字数: 152 千字

版 次: 1998 年 12 月第 1 版 1998 年 12 月第 1 次印刷

书 号: ISBN 7-302-03271-8/TP·1750

印 数: 0001~5000

定 价: 13.00 元

序 言

近年来 Internet 以迅猛之势发展,已成为全世界规模最大的计算机网络,网上资源丰富,为世界各国千万用户所瞩目,我国也已于 1994 年 5 月正式入网,并已有成千上万使用者上网工作,如何充分、合理地利用 Internet 的网络与信息资源,为我国社会的各方面服务,是我们应予以重视和研究的问题。

Java 的出现正迎合了 Internet 发展形势的需要,它所体现的简单、不依附于平台、面向对象、分布式、可靠性、安全性、可移植性、动态性、多线程等特性,为 Internet 的使用提供了一种良好的开发和运行环境,成为 Internet 适用、新型的编程语言。Java 出现后立即为世界各大公司所注目,纷纷购买 Java 使用权,并在剧烈的市场竞争中展开了大规模的研究与产品开发。Java 语言及其相关技术同样也引起了我国学术界、产业界和应用界的广泛关注和兴趣。因此 Java 丛书的出版发行,是符合时宜之举,必将受到大家的欢迎。

根据编著者的计划,Java 系列丛书将包括:《Java 语言入门》,《Java 虚拟机规范》,《Java 语言编程技术》,《Java 语言 Applet 编程技术》,《Java 语言 API 类库》,《Java 软件包的使用》,《Java 语言 SQL 接口》,《Java 语言调试技术》及《HotJava 使用指南》,《Java 嵌入技术》等十多部书,达几百万字。详细深入地介绍 Java 语言及其相关技术并在短期内出齐 Java 系列丛书,为我国广大读者研究和掌握 Java 提供一种系统而全面的文献资料,无疑是一种十分有意义的事。Java 作为新生事物,尚在不断发展完善之中,因此我希望我们对 Java 及其相关技术不只停留于被动的学习、跟踪与使用,还应以积极主动的态度,通过应用实践和深入分析研究,参与开发创新,为计算机的网络应用做出我们自己的贡献。

清华大学计算机科学与技术系王克宏教授及其知识工程科研组,在 Internet 上进行多层次的研究开发工作已有一段时间,取得了阶段性成果,并曾在不久前召开的“中国计算机学会第九次全国学术大会”的全体会议上作报告,受到与会者的欢迎。相信他们在结合自己研究工作基础上编写的这套 Java 系列丛书,必将有助于我国信息科学技术的发展。我高兴地为此丛书作序并祝他们成功!

张汝祥

1996.8

* 本文作者为中国科学院院士、中国计算机学会理事长

开辟 Java 计算的新时代

Sun 公司发明的 Java 语言风靡全球。据 Forester Research 公司对《幸福》杂志评出的世界 1000 家大型企业进行的调查,目前在它们中间已有 62% 正在使用 Java 从事开发工作;有 42% 已将 Java 纳入自己企业一年内的战略开发计划。这个调查结果表明,Java 已为大家所接受。

计算机自从 50 年前诞生以来,经历了几个发展时代:终端/主机计算、PC 计算、客户机/服务器计算,而现在已发展到一个新的阶段,即 Java 计算时代。这个时代的特点就是以 Java 为代表的网络计算。

Java 带来的是一场革命。这是第一个真正独立于平台的计算方案,它能充分发挥 Internet 的作用。Java 计算是实现“一个世界,一个网络”构想的关键。这样一个透明的、全球连接的和信息交换的网络,可将所有最新的计算技术、电话、出版/媒体和娱乐集成一体。过去妨碍这一构想成为现实的是计算平台不能兼容,而 Java 语言却以其许多优秀特性使之成为允许各类系统相互兼容和共享应用环境的桥梁。这使各类软件“一次写成,到处可用(write once/run anywhere)”,这样,相同的软件可在不同器件上运行,无论是 PC 机、苹果机、UNIX 计算机,还是顶置盒、PDA(个人数字助理)、移动电话,乃至智能元器件,无一例外。Java 的这一贡献将使全人类受益,因而给人们的生活方式带来极大的改变。

清华大学王克宏教授是最早研究 Java 语言的中国学者之一。在他的周围,许多热心于 Java 计算的青年学者、博士生、硕士生和本科生组成了一支朝气蓬勃、积极进取的生力军,他们思想活跃、工作勤奋、通过努力把 Java 语言的开发和应用提高到一个新的水平。据我所知,在中国的诸多大专院校和科研院所,包括北京航空航天大学、上海复旦大学,也都有许多学者致力于 Java 的研究和开发。由于他们的辛勤努力和积极探求,在 Java 平台上开发和使用的 Java 应用将给中国的信息业带来新的发展,使 Java 计算在中国进一步得到推广。

《Java 系列丛书》就是 Java 开发园中的一朵奇葩,它不倜傥,以其朴实无华、实用有效的风格展示于人。我们希望由王克宏教授主编的该套丛书能赢得广大读者的喜爱,希望能有更多的中国智者和用户在 Java 计算的时代熟悉 Java,掌握 Java,利用 Java 的诸多优点,编写出更多更美的应用软件,从而造福于人类。

谨以此愿祝贺 Java 系列丛书的出版发行。

李永超

1996.10

* 本文作者为 Sun 公司大中国地区副总裁

前 言

在当今的信息化浪潮中,不仅有计算网络化的趋势,还有着计算嵌入化的趋势。Java 源于嵌入计算,成长于网络计算。在这两个趋势关系日益密切的情况下,Java 的地位日益加强。要全面了解 Java,就必须了解 Java 嵌入技术。

通过本书,读者将了解和掌握 Java 嵌入技术。读者将了解和掌握 Personal Java, Embedded Java, Java Card 规范并学会相应的编程。读者还可以详细地了解 Java 芯片的结构、特点和应用。需要强调的是,本书目的不仅仅在于介绍技术本身,本书还力求使读者对 Java 嵌入技术在整个信息化浪潮中的广阔的应用前景和不可代替的地位有一个较深刻的理解。

因此,本书的读者有一个较广的范围。具体说有以下两类:

1. 技术人员和对 Java 嵌入技术感兴趣的师生。本书作者收集和综合了完稿时几乎所有公开的与 Java 嵌入技术相关的技术资料。虽然现在技术发展很快,也许你拿到这本书时,书中有些内容已经有些过时,但你所了解和掌握的一些基本概念和思想也许不会过时。

2. 对 Java 嵌入技术感兴趣但不想也不需要掌握细节的信息产业人士。本书对 Java 嵌入技术对整个信息化浪潮的影响作了分析,并介绍了 Java 嵌入技术的广阔的应用前景。

本书分六章,它们是:

第一章,概述 本章概述了 Java 和 Java 嵌入技术的历史,它还作为全书的知识铺垫,简要地介绍了 Java 语言的特点,Java 虚拟机和 Java 操作系统。

第二章,Personal Java 本章探讨了 Personal Java 的特点与 Personal Java 平台的结构并说明如何用 Personal Java 编程,简要介绍了目前尚未公布详细技术资料的 Embedded Java。本章还简要介绍了与 Personal Java 或 Embedded Java 密切相关的用于嵌入平台的实时操作系统 CHORUS/OS 和浏览器 Personal WebAccess。

第三章,Java 智能卡 本章介绍了 Java 智能卡系统的结构特点及它与 Personal Java 和 Embedded Java 的联系及区别。

第四章,Java 智能卡编程 本章详尽地阐述了在与计算机环境差别很大的智能卡环境中如何用 Java 进行编程。

第五章,Java 芯片结构 本章介绍了 Java 芯片的结构特点和其指令系统。

第六章,Java 嵌入技术应用前景 本章先总结了前几章的内容,然后探讨了嵌入技术应用的广阔前景和 Java 在嵌入技术中的优势。本章还介绍了 Web 电话等目前已有的 Java 嵌入技术应用实例。最后,本章还阐述了由作者提出的嵌入式瘦客户机概念并探讨了它对信息化浪潮可能的影响。

读者或许并不一定需要读完整本书,对于不同的读者,作者有以下的建议:

1. 如果你并非技术人员,建议你把阅读重点放在第一章的第一节、第二节以及最后一章。

2. 如果你的目标只是掌握 Personal Java 编程,建议阅读重点放在第一章、第二章和第六章。第五章也应该看一下。

3. 如果你的目标只是掌握 Java Card 编程,建议阅读重点放在第一章、第三章、第四章、第六章。第五章也应该看一下。

4. 如果你对整个 Java 嵌入技术都兴趣十足,那么读完整本书吧!

还应强调的是,本书的各章都有相对的独立性,不一定要按顺序读,比如你可以读完第一章后就读第六章。不过,一开始最好先读一下第一章。此外,第三章和第四章的先后阅读顺序也不要变。

本书的第一章的第四节和第四章是在徐鹏同学的文章的基础上改写的,Sun 公司的高克家先生对本书探讨的嵌入式瘦客户机概念提出了很好的意见,Sun 公司的范百炼先生为本书的资料收集提供了热心的帮助,对此一并表示感谢。

本书作者水平有限,诚恳地希望读者对书中的缺点提出批评,批评意见请发往 xjj@263.net 或 j94@keg.cs.tsinghua.edu.cn。

目 录

第一章 概 述	1
1.1 Java 嵌入技术的历史	1
1.2 Java 语言特点	2
1.3 Java 虚拟机简介	4
1.4 Java 操作系统	7
第二章 Personal Java	12
2.1 概 述	12
2.2 Personal Java AWT	17
2.3 图形处理	23
2.4 降低内存需求	28
2.5 Personal Java 的其他特点	31
2.6 Embedded Java 简述	34
2.7 Personal Java 开发工具及开发过程	35
2.8 JavaCheck 的使用	37
2.9 Personal WebAccess——Personal Java 平台的浏览器	40
2.10 CHORUS/OS	41
第三章 Java 智能卡	44
3.1 概 述	44
3.2 精简后的 Java 语言	46
3.3 精简后的 Java 虚拟机	49
3.4 例 外	51
第四章 Java 智能卡编程	53
4.1 Applet 程序的一些基本概念	53
4.2 Applet 程序设计	58
4.3 APDU 的处理	62
4.4 一些常用的类	68
4.5 文件系统	73
4.6 一个例子	84

第五章	Java 芯片	89
5.1	概 述	89
5.2	RISC-Like 的结构	90
5.3	革新的堆栈结构	92
5.4	组成单元	94
5.5	内存控制	95
5.6	picoJava 芯片内部信号	95
5.7	microJava701 简介	97
第六章	Java 嵌入技术的应用前景	99
6.1	四个平台的关系	99
6.2	嵌入技术的广阔前景	101
6.3	Java 嵌入技术的优势	103
6.4	Java 嵌入技术的应用实例	104
6.5	嵌入式瘦客户机与 Java	106

第一章 概 述

1.1 Java 嵌入技术的历史

要说 Java 嵌入技术,就要先说一下什么是嵌入技术。嵌入技术是一个简化的说法,严格说应该是嵌入计算技术。现在计算机技术发展很快,很多词的涵义也在变,因而嵌入计算这个词的含义也不止一个,本书所指的是目前最为普遍的含义,即指把计算机引入各种嵌入设备,如电视、电话、数控机床、PDA(个人数字助理)等,从而使后者具有可编程的特性。因此,嵌入应用需要微处理器,在大多数情况下需要操作系统,并且操作系统一般是实时操作系统 RTOS,如本书将介绍的 CHORUS/OS,这种操作系统与一般计算机常见的操作系统如 Windows95/NT,UNIX 等是很不一样的,关于 CHORUS/OS 本书的第二章还有进一步的介绍。

相信大多数读者在看本书前一定对 Java 已经有了一个直观的了解,并且这种直观的了解一定来自 Java 在 Internet 上的应用。但本书说的是 Java 嵌入技术,这与你们以前所知道的,所理解的 Java 是不太一样的。Java 在 Internet 和 Web 中的地位是显然的,可以说是它们的首选语言,但 Internet 应用环境和嵌入环境是不一样的,为什么 Java 也可以适用于嵌入应用中呢?要说明这一点,需要从 Java 的起源说起。

80 年代末 90 年代初,有一种说法,就是在 90 年代,嵌入计算应用尤其是消费电子设备中的嵌入计算应用,将会有很大的发展。在 80 年代,先是苹果公司的 Apple 微机,后是 IBM 的 PC 及其兼容系列微机取得了相当大的成功。微机的这种成功的根本原因就是随着微电子技术的发展,CPU 的性能越来越高,存储器的容量越来越大,同时它们的价格却越来越低廉。微机的这种成功使人们不禁想到随着技术的进一步发展,还有比微机更小的“计算机”也将获得成功,这就是各种嵌入设备尤其是消费式电子设备。这种说法不见得完全正确(这一点本书的最后一章还会做进一步的讨论),但在当时却引起了一些公司的注意。Sun 公司就是其中的一个。

在 90 年代初,Sun 在 UNIX 工作站市场已取得了相当大的成功,但 Sun 公司却不满足于此,它把目标放在了其他与计算机技术相关的领域。1991 年,正是看到了嵌入计算应用市场的广阔前景,Sun 公司的由 James Gosling, Bill Joe 等人组成的 Green 小组开发了一个名为 Oak 的软件(这是 Java 的前身),开发它的目的是用于电视等家用电器的嵌入式应用。但后来,正如我们现在知道的,Java 的发展却出乎它的创造者的预料,它并没有在嵌入应用中大展身手(起码,目前是这样),却风靡于 WWW 世界。现在,人们提到 Java 总是联想到 Web 和 Internet。

1995 年 5 月,Sun 在 SunWorld 上正式发布了 Java 和 HotJava 浏览器。1996 年 Java 赢得业界的大力支持,Java 应用软件纷纷面世,Java 被评为当年的十大科技成果之一。1997 年 JDK1.1 发布,国际标准化组织批准 Sun 公司作为 Java 公开有效规范(PAS)提案者的申请。Java 的这一系列发展,似乎都只与 Internet 和 Web 相关。但是 Sun 公司在做出了把原先目标在嵌入应用的 Java 转而用于 Internet 和 Web 的决定并取得了巨大的成功之后,并没有放弃

其在嵌入计算中的应用。正好相反, Sun 一直声称 Java 不但要用于 Internet, 还可以在嵌入应用中大有作为。笔者现在还记得在自己 1996 年刚接触 Java 时, 总是看到 Sun 公布的资料中说什么不但浏览器中有 Java, 微波炉中也会有 Java, 这让当时的我感到极大的兴趣——微波炉中怎么会有 Java 呢? 有了 Java 又会有什么用呢? 希望读者也有很大的兴趣关心这样的问题, 这将使你饶有兴趣地看这本书。

Sun 公司在大力开发 Java 在 Internet 中的应用技术时, 丝毫没有放松 Java 嵌入技术的开发。早在 1996 年初, Sun 公司就公布了 Java 芯片的初步资料, 同年底公布了 JavaCard 1.0。1997 年底又正式公布了 PersonalJava 和 JavaCard 2.0 并兼并了从事嵌入式实时操作系统开发的 Chorus 公司。1998 年初公布了 EmbeddedJava 的初步资料。就在本书完稿时, 各项 Java 嵌入技术的开发仍在积极进行中; Sun 正在为 1998 年底 MicroJava701 芯片的大批量生产做准备; Gemplus 正在开发采用 picoJava 核心的 32 位新一代智能卡; PersonalJava 正准备运用于 Windows CE 中……

如今, 一股关于 Java 应用新的浪潮正悄然兴起。这就是本书将介绍给你的 Java 嵌入技术。图 1.1 是三星公司的与 PersonalJava 平台兼容的 Web 电话。这种电话带了一个简单的显示器和一个键盘, 它可以收发电子邮件, 浏览万维网等。



图 1.1 三星公司的 Web 电话

1.2 Java 语言特点

相信读者中有不少人对 Java 语言本身已经相当了解了(如果你就是这种读者, 那么可以不用读本节), 但笔者希望本书有一个较广的读者范围, 所以仍需介绍一下 Java 语言的特点。

Java 如此广受欢迎不是像有些人说的那样纯粹是商业宣传的结果, 而是有其内在的原因。下面就是其特点:

1. 简单

有人说从语法形式上讲, Java 语言是一种简化的 C++, 这种说法不无道理。Java 略去了 C++ 中的繁琐的运算符重载, 多重继承等模糊的概念, 并且通过实现自动垃圾回收大大简化了程序设计者的内存管理工作且减少了在内存管理上出错的可能。作为一种面向对象语言, Java 通过提供最基本的方法来完成指定的任务, 只需理解一些基本的概念, 就可以用它编出适合于各种情况的应用程序。

2. 短小

Java 不但语法形式简单,实现起来也简单。它的基本解释器及其类的支持只有 40KB 左右。加上标准类库和线程的支持也只有 200KB 多。一个同样的程序,用 Java 一般要比 C++ 等语言短小得多(指编译后)。出现这种情况的原因是 Java 最初准备用于资源极为有限的嵌入设备,开发它时特别注意它的可执行代码的短小,比如 Java 虚拟机以堆栈机为中心,省去了大量的内存操作,同时 Java 的每条字节码都只有一字节,这些都使得 Java 可以在十分有限的系统资源下得以很好地实现,这一点对嵌入应用很有价值。

3. 面向对象

Java 语言的设计集中于对象及其接口,它提供了简单的类机制以及动态的接口模型。对象中封装了它的状态变量以及相应的方法,实现了模块化和信息隐藏。而类提供了相应对象的原形,并且通过继承机制,子类可以使用父类所提供的方法,实现了代码的复用。代码复用是嵌入计算编程一直追求的东西。

4. 取消指针

没有指针的高级语言不只 Java,但 Java 与 FORTRAN77 这样的没有指针的语言不同,它是在可以完成有指针的高级语言所能实现的各种功能的情况下“取消”了指针。Java 中的引用概念使得它可以在没有指针的情况下实现对数据结构的各种复杂操作,Java 的内存垃圾自动回收使编程人员在没有指针的情况下照样可以高效地使用内存。

笔者认为,在面向对象时代,指针已无存在的必要。C++ 因必须和 C 语言保持兼容不可能取消指针,但 Java 做到了。读者在学习 C 语言或 Pascal 时可能被告知指针是如何的有用,但技术的发展实在是太快了,指针因其太易出错在面向对象时代可以被取消。从没有指针到有指针再到没有指针,经过了一个否定之否定的过程,Java 语言是这一发展过程的结果。

5. 平台无关

关于 Java 的平台无关性也许有的人有一种误解,把 Java 的平台无关性归结于它与 BASIC 等语言一样是解释执行的语言,这种说法并非完全正确。Java 与 BASIC 这样的传统的解释执行语言是不同的。任何 Java 程序都被编译成 Java 字节码,Java 字节码与其他一些程序执行信息组成了 Java 类文件(.class 文件)。字节码可以被看成与汇编语言类似的东西,但是它经过特殊设计特别适用于用软件解释的方式加以执行。

由于 Java 字节码可以被软件解释执行,这样只要安装了 Java 运行的软件系统,Java 程序就可以在任意的处理器上运行。这些字节码指令对应于 Java 虚拟机中的表示,Java 解释器得到字节码后,对它进行转换,使之可以在不同的平台上运行。当 Java 字节码被软件解释器解释执行时,字节码本身携带了许多编译时的信息,使得连接过程更加简单。

对于类似汇编语言的 Java 字节码,只要稍加扩充,就可以成为某种微处理器的指令系统。这种微处理器就是 Java 芯片,本书第五章将详细介绍。

6. 鲁棒性

Java 在编译和运行程序时,都要对可能出现的问题进行检查,以消除错误的产生。它提供自动垃圾收集来进行内存管理,防止程序员在管理内存时容易产生的错误。通过集成的面向对象的例外处理机制,在编译时,Java 提示出可能出现但未被处理的例外,帮助程序员正确地进行选择以防止系统的崩溃。另外,Java 在编译时还可捕获类型声明中的许多常见

错误,防止动态运行时不匹配问题的出现。

7. 分布性

Java 是面向网络的语言,通过它提供的类库可以处理 TCP/IP 协议,用户往往可以通过 URL 地址在网络上很方便地访问其他对象。读者也许会问,本书说的是 Java 嵌入技术,还说什么分布性呢?实际上不能把 Java 在嵌入设备中的应用和其在网络上的应用分割开来,很多嵌入应用都有联网要求,如 Web 电话,顶置盒等。关于计算网络化和计算嵌入化的关系问题,本书后面的章节尤其是最后一章还有进一步的探讨。

8. 多线程

多线程机制使应用程序能够并行执行,而且同步机制保证了对共享数据的正确操作。通过使用多线程,程序设计者可以分别用不同的线程完成特定的行为,而不需要采用全局的事件循环机制,这样就很容易地实现网络上的实时交互行为,同时对嵌入设备的实时控制也很有好处。

9. 动态性

Java 的设计使它适合于一个不断发展的环境。在类库中可以自由地加入新的方法和实例变量而不会影响用户程序的执行。Java 通过接口来支持多重继承,这样做确实有些不足,但也使之比严格的类继承具有更灵活的方式和扩展性。

10. 可移植性

与平台无关的特性使 Java 程序可以方便地运行在网络上的不同计算机和不同的嵌入设备。但这同时也要求 Java 的运行系统可以移植到各种硬件环境,实际上 Java 的运行系统和很多开发工具本身的很大一部分是用 Java 语言写的,剩下的部分也是用标准 C 语言实现,这使得 Java 系统本身也具有较强的可移植性。本书中介绍的 Java 嵌入应用开发工具也有一些自身是用 Java 写成的。

11. 安全性

大家知道,对于网络环境中的 Java 来说,安全性是很重要的,必须确保病毒不会通过网络上的 Java 程序来进行传播。实际上,嵌入应用也要求 Java 具有很好的安全性,比如智能卡,一旦安全性出了问题,后果不堪设想。

幸运的是 Java 天生具有完备的安全保证机制。首先,Java 语言不支持指针,一切对内存的访问都必须通过对象的实例变量来实现,这样就防止了程序员使用“特洛伊”木马等欺骗手段访问对象的所有成员,同时也避免了指针操作中容易产生的错误对安全的破坏。其次,Java 还有严格的安全性检查,一是在编译时就有安全性检查;二是在解释 Java 字节码时有严格的安全检查,一些完好的通过编译的 Java 程序如“图谋不轨”照样会被禁止执行。

1.3 Java 虚拟机简介

1.3.1 什么是 Java 虚拟机

读者在本书以后的章节中将大量接触到 Java 虚拟机的概念。这是因为本书不是仅仅介绍 Java 嵌入技术的编程,而是想把 Java 运用到嵌入设备中时的整个系统构成做一些介绍。这就必然要经常讲到 Java 虚拟机。要想较好地弄明白本书所有内容就起码需要明白

Java 虚拟机是什么。Java 虚拟机是什么？这就是本小节要说的。

Java 虚拟机就是用来运行经过编译的 Java 字节码的计算机实现,这种实现既可以是软件实现也可以是硬件实现。从 Java 应用程序角度来看,Java 虚拟机是一个假想的计算机平台,Java 应用程序运行在 Java 虚拟机这个假想计算机之上并通过它获得所需的本地资源。从计算机或嵌入设备角度来看,Java 虚拟机是建立在处理器等硬件设备基础之上的“上层建筑”,在本设备上运行的所有的 Java 应用都通过它执行。

Java 虚拟机的实现方案有两种。一是纯软件仿真,正是大家目前所接触的形式;其二就是用 Java 芯片了,但用 Java 芯片不是说虚拟机中就没有软件组成了,仍然有一部分需要用软件实现。

要了解 Java 虚拟机,就必须了解 Java 程序生成实际机器可执行代码的过程。许多程序设计语言通过源程序进行编译和链接,直接生成可执行的代码。而各种 Java 程序,无论是标准的 Java 程序,还是本书将介绍的专门用于嵌入应用的 Java 程序则和它们不同,Java 应用程序的这个过程包括程序的编译,字节码的装入,校验,解释或编译。关于 Java 程序执行的这一系列过程见图 1.2。

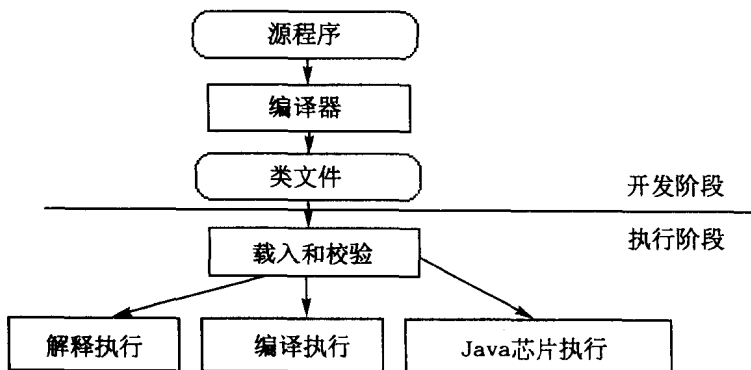


图 1.2 Java 程序生成实际机器可执行代码的过程

首先是对 Java 程序的编译,以生成字节码。这个过程由程序开发人员用的 Java 编译器完成。前面说过,字节码的层次相当于汇编语言一级。但是它经过了专门的设计以有利于软件仿真的方式实现,并不针对某种特定的计算机硬件平台。需要注意的是,在这一步中,对变量和方法的引用,不是确定为数值引用,即通过具体的偏移量的引用,而是将符号引用的信息保存在字节码中。

通过编译后,就意味着完成了 Java 程序的开发,接下来就是 Java 程序的执行阶段了。

在执行阶段中,先是进行字节码的装入,校验,由本地计算环境上的相应软件完成——无论是否用 Java 芯片。类装入器装入程序的所有代码,包括程序中调用,包含和继承的所有类的代码。每个类都被装入一个独立的名字空间内,彼此之间只有通过符号引用才能相互作用。本地的类和外部的类在地址空间上是分开的。所有的类都装入以后,可执行代码的内存布局就被确定。由符号引用到内存地址空间的查询表也建立起来了。然后字节码校验器对装入的字节码进行校验,以排除错误和不安全的因素,如果发现了安全性上的问题,将退出 Java 程序的执行过程。

在通过了装入和校验之后,最后就是 Java 字节码的运行了。这时有三种方式,一是传

统的方法,即由解释器解释执行。但是解释执行速度慢,于是有人发明了第二种方式——编译执行。要说明的是,这时的编译的含义与通常所说的编译不一样,实际上 Java 程序在准备执行前已经通过了编译(通常意义上的编译)成了字节码,但执行时的编译指的是把 Java 字节码一次性地翻成本地计算环境的机器代码然后执行,而不是像解释器那样边解释边执行。第三种方式也是最快的一种方式就是用 Java 芯片直接执行了,这种方式不但速度快,并且省内存,因为它省去了一个软件实现的解释器或编译器。

1.3.2 Java 虚拟机与平台无关性

Java 虚拟机是 Java 的平台无关特性得以实现的关键。如图 1.3 所示,Java 平台结构中 Java 虚拟机起着承上启下的作用。

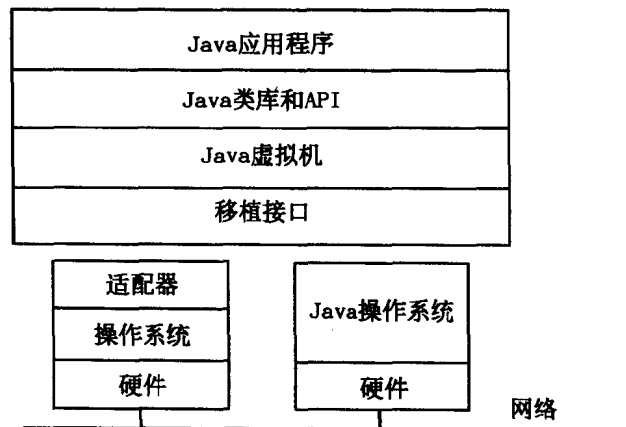


图 1.3 Java 虚拟机与 Java 平台

在 Java 虚拟机的下方是移植接口,移植接口由依赖于平台的和不依赖于平台的两部分组成,其中依赖于平台的那部分为适配器。Java 虚拟机通过移植接口在具体的操作系统上实现。如果是在 Java 操作系统上实现,则不需要依赖于平台的适配器,因为这部分工作已由 Java 操作系统完成。因此,对于 Java 虚拟机来说,操作系统和更底层的硬件是透明的,即对于 Java 虚拟机这一层次来说,操作系统和硬件好像不存在一样,因此也无需考虑。

在 Java 虚拟机上方是 Java 应用程序接口,它由 Java 基本类和 Java 标准扩展类组成。我们编写的 Java 应用程序就运行在 Java 的类库和 API 上,因而对于它们来说,操作系统和硬件就更是透明的了。我们编写的 Java 程序因此可以在任何 Java 平台上运行而无需修改。

Java 虚拟机定义了独立于平台的类文件格式和字节码形式的指令集。在任何 Java 程序的字节码表示中,变量和方法的引用都是使用符号,而不是使用具体的数字。用数字引用替代符号引用是在运行时由解释器完成的。当第一次引用一个变量或方法的时候,用它的名字在查找表中查找,以确定它的数字偏移量。这种查找和替换只在第一次引用时需要,因此对速度影响不是很大。由于存储器的布局要在运行时才确定,所以对类的变量和方法的改变不会影响现存的字节码。例如,某个 Java 程序引用了其他系统中的某个类,该系统中那个类的更新不会使你的程序崩溃。如果是 C++,你必须将你的程序中相关部分再次编译。Java 的这种特点不仅提高了 Java 的平台独立性,同时也使得嵌入应用中的远程控制成为

可能。

1.3.3 Java 虚拟机与安全性

前面说过,无论是对于网络环境,还是嵌入计算环境,安全性是必须要求的。Java 虚拟机在 Java 的安全体系中占有重要地位。至少有以下三项安全性是与 Java 虚拟机相关的。

其一是内存的布局。我们说过,在任何 Java 程序的字节码表示中,变量和方法的引用都是使用符号,而不是使用具体的数字。用数字代替符号引用的工作由解释器或编译器在运行时完成。对于编程人员来说,内存分配是透明的,不可能在编写程序的时候就知道内存的布局。编程人员无法通过伪造指针来访问内存空间。

其二是字节码检查。虽然安全性检查在 Java 源程序编译成字节码时就已经进行过,但这还不够。当从其他地方下载 Java 字节码时,并不能认为它是安全的。因为有可能是生成这些字节码的编译器本身有问题,比如有人故意做了一个不合规范的 Java 编译器,以侵入其他用户的系统。所以必须对字节码进行检验。字节码的一个重要性是它的可解析性很强,能够对指令集进行分析,并对其以后的行为做出一定的推断,这就为检验打下了基础。通过检验的程序,可以确保不存在伪造的指针,不违反访问权限,不非法访问对象,不会导致操作数栈溢出等违反系统安全性的操作。

其三是字节码装载器的安全检查。因为字节码采用符号引用,内存布局在运行时确定,所以可以在字节码装入时将本地的类和外来的类严格区别开来。本地的类共有一个统一的名字空间。这种机制为本地的类建立起一道安全屏障。

虽然标准的 Java 虚拟机已有很高的安全性,但嵌入设备产商还不满足。法国的一家智能卡产商开发了“安全 Java 虚拟机”SJVM。这是一种用于对安全性要求特别高的嵌入设备中的 Java 虚拟机,它在标准的 Java 虚拟机的基础上,又进一步增加了强大的安全保护机制,不但在字节码载入时有安全检查,在执行时还对 Java 字节码边执行边检查,确保不执行破坏操作。

1.4 Java 操作系统

1.4.1 什么是 Java 操作系统?

读者一般以前见过“JavaOS”这个词。首先要说明的是,严格来说“JavaOS”和“Java 操作系统”这两个词并不是完全相同的。顾名思义,Java 操作系统指的是专门为运行 Java 应用程序设计的操作系统。而 JavaOS 则是 Sun 公司的一个产品的名称,是一种 Java 操作系统。除了 Sun 开发的 Java 操作系统之外,其他公司如 IBM,日立等也在开发 Java 操作系统,如 JavaOS 等,你是不是想起了“金庸”和“金庸”?

本书讲的是 Java 嵌入技术,为什么要介绍 Java 操作系统呢?原因很简单,嵌入应用中用到 Java 操作系统的可能性要比在计算机中用到它的可能性来得大。在计算机领域,没有 JavaOS,我们照样可以在其他操作系统上用软件实现的 Java 虚拟机运行 Java 应用程序。在嵌入计算领域,系统资源十分有限,我们虽然也可以在任一嵌入设备用的操作系统上用软件实现的 Java 虚拟机来运行 Java 应用程序,但这会消耗宝贵的系统资源,这时用 Java 操作系

统就很有必要了。此外,嵌入应用中 Java 操作系统广阔的应用前景也和信息产业的现状有关,这一点本书最后一章再做详细说明。

虽然 Java 操作系统不仅仅指 JavaOS,但是 JavaOS 是目前最有前途的 Java 操作系统。本书以它为例来介绍一下 Java 操作系统。

Java OS™是为使 Java™能够在各种计算机和嵌入设备平台上运行优化而设计的一种新的平台。为使 Java 应用程序能够直接在硬件平台上运行,而不需要一个像 UNIX, Windows 95/NT 这样的主操作系统,JavaOS 特别提供了一种运行时环境(runtime)。JavaOS 提供 Java 平台以自动运行利用 Java 语言设计的功能强大的 Applet 程序及其他应用程序。同样地,它亦提供 Java 虚拟机和低层功能以实现窗口、网络以及文件系统功能,而无需任何主操作系统的支持。

JavaOS 由两部分组成。一是本地代码部分,具有特定的指令集和硬件平台;二是 Java 代码部分。其中 Java 代码具有平台独立性。JavaOS 定义的平台包括 CPU、物理内存、其他附加设备、总线和插槽等构件。操作系统的平台独立性组件被称为 Java 运行时环境,而操作系统中依靠平台的部分则可称为 JavaOS 内核。JavaOS 的组成见图 1.4。

经过设计 JavaOS 可以在一台硬件条件非常有限的平台上运行,这对嵌入应用很重要。例如,运行时本身既不需要内存管理单元 MMU(Memory Management Unit)把虚拟地址映象到物理内存地址,又不需要内存保护。而是由其底层的内核决定是否使用 MMU 或强制执行内存保护。再例如,JavaOS 的内核使用内存管理单元 MMU 来建立一些接近 Java 运行时环境但相互隔离的物理内存区域。未来的 JavaOS 也许还能以一种更为积极的方式来管理 MMU。

JavaOS 被建构成一个层状结构,每一层都以独立的形式存在。这种结构可以满足两种需求:即产品定制工作和一个并行操作系统的模型。

基于产品定制的 JavaOS 就是为了满足产品的限定和要求而集合 JavaOS 的几个层。比如说,运行 JavaOS 的智能型电话需要一个实时内核、Java 虚拟机、最基本的图形能力和一些通讯协议。而一台网络计算机则需要一个没有实时约束但拥有更多也更完整特性的内核、Java 虚拟机、抽象窗口工具集 AWT(包含完整的图形库)以及所有其他的在 Java 开发包 JDK 之内的应用程序接口 API 和 HotJava™浏览器。

图 1.4 展示了在不具备一种主操作系统的前提下运行 Java 平台时所使用的软件体系结构,Java 应用程序接口 API 之上的程序是具有平台独立性的 Java 应用程序和 Applet 程序。

JavaOS 是否真的是一个操作系统?这完全依赖于用户自己的观点。JavaOS 与传统的操作系统在许多方面截然不同,这些方面包括:

- 需要一个文件系统;
- 需要一个虚拟存储器;
- 需要相互独立的地址空间;
- 支持多种程序语言;
- 有自己的系统调用设置。

JavaOS 与一个传统的操作系统在一些方面亦有类似之处,这些方面是:

- 可以自引导;
- 支持口令保护的登录特性;
- 可以安全地同时运行几个 Applet 程序;