

游戏开发经典丛书

游戏设计天才扛鼎巨献

游戏脚本设计技术全面掌握

涉及 Lua、Python、汇编语言及
高级语音解析器

更是一本学习编译原理的好书

随书
附赠



1CD

Game Scripting Mastery

游戏脚本 高级编程

[美] Alex Varanese 著

陈 洪 罗颖民 李华杰 译

北京递归开元教育有限公司 审校

游戏开发经典丛书

游戏脚本高级编程

(Game Scripting Mastery)

(美) Alex Varanese 著

陈 洪 罗颖民 李华杰 翻译

北京递归开元教育有限公司 审校

清华大学出版社

北 京

内 容 简 介

本书引领读者进入程序员们称之为“游戏脚本编程”的全新领域,本书将从概括论述什么是脚本编程以及它是如何实现的开始,阐述游戏编程的理念,根据作者多年的开发经验,使读者掌握使用脚本是把游戏代码和主引擎分离的最理想的办法,带领读者进入神秘的游戏脚本语言世界,学习如何编写脚本语言,编译器理论,享受游戏编程方面的极大乐趣。

本书是游戏开发经典丛书系列之一,适合游戏开发人员、业余游戏软件开发爱好者,也可以作为大专院校相关专业的参考书。

Alex Varanese

Game Scripting Mastery First Edition

EISBN: 1-931841-57-8

Copyright 2004 by The Premier Press, Inc. a division of Thomson Learning.

Original language published by Thomson Learning (a division of Thomson Learning Asia Pte Ltd). All Rights reserved.

本书原版由汤姆森学习出版集团出版。版权所有,盗印必究。

Tsinghua University Press is authorized by Thomson Learning to publish and distribute exclusively this Simplified Chinese edition. This edition is authorized for sale in the People's Republic of China only (excluding Hong Kong, Macao SAR and Taiwan). Unauthorized export of this edition is a violation of the Copyright Act. No part of this publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

本中文简体字翻译版由汤姆森学习出版集团授权清华大学出版社独家出版发行。此版本仅限在中华人民共和国境内(不包括中国香港、澳门特别行政区及中国台湾地区)销售。未经授权的本书出口将被视为违反版权法的行为。未经出版者预先书面许可,不得以任何方式复制或发行本书的任何部分。

北京市版权局著作权合同登记号 图字:01-2005-0941

本书封面贴有汤姆森学习出版集团激光防伪标签,无标签者不得销售。

978-981-4195-73-7

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

图书在版编目(CIP)数据

游戏脚本高级编程/(美)瓦兰尼斯(Varanese, A.)著;陈洪,罗颖民,李华杰译.——北京:清华大学出版社,2006.10
(游戏开发经典丛书)

书名原文:Game Scripting Mastery

ISBN 7-302-13079-5

I. 游… II. ①瓦… ②陈… ③罗… ④李… III. 游戏—应用程序—程序设计 IV. ①G899 ②TP311

中国版本图书馆CIP数据核字(2006)第051683号

出 版 者:清华大学出版社

<http://www.tup.com.cn>

社 总 机:010-62770175

组稿编辑:高巍

封面设计:范华明

印 刷 者:北京密云胶印厂

发 行 者:新华书店总店北京发行所

版 次:2006年10月第1版 2006年10月第1次印刷

书 号:ISBN 7-302-13079-5/TP·8290

印 数:1~4000

定 价:98.00元(附光盘1张)

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

文稿编辑:马子杰

版式设计:王慧娟

装 订 者:三河市金元印装有限公司

开 本:185×260 印张:57.25 字数:1284千字

前言

游戏编程是很有意思的！原因在于你在一个系统里需要编写许多种不同的子系统，而不用管它是否是一个简单的派克人（Pac Man）复制或者一个复杂的 triple-A 战术射手。同时，编程经验都是极为丰富的，不管你在编写渲染程序，还是声音系统、AI 系统，或者游戏本身的代码，所有这些类型的编程都包含着你将要解决的挑战。要在任何一个领域中编写代码的最好办法就是提前尽可能地学习这方面的知识，这也就是为什么你需要在手边摆上一大堆的编程书籍的原因。

游戏编程现在很少涉及的一个领域就是脚本。有些游戏不需要脚本——游戏是否需要脚本常常依赖于你的开发环境和开发组——但是在大多数情况下，使用脚本是把游戏代码和主引擎分离开的最理想的办法，甚至在处理游戏中的动画时。大多数编程者当要解决一个特殊的编码问题（例如处理 NPC 交互）时，通常会决定编写他们自己精细定制的与他们的游戏代码集成在一起的语言。今天已经有很多可用的脚本工具，所以要做到这一点并不难，而且很有趣！

许多编程人员现在还没有意识到脚本解决方案的可用之处，这也就是本书中要介绍的内容。《游戏脚本高级编程》将会带你进入神秘的游戏脚本语言世界。你将会学到什么是脚本语言以及如何来编写脚本语言；你将会学习 Lua、Python 和 Tcl 以及如何在你的游戏中使用这些脚本语言（顺便提一句，我是支持 Lua 的中坚分子）；当然你还会学到编译器理论。你甚至会看到如何开发一个完整的脚本语言！这里有很多的知识，如果你热爱编写游戏，我相信你会欣喜地找到游戏编程方面的更多东西。慢慢品味吧！



John Romero

致 谢

一切都开始于 2001 年在加利福尼亚圣克拉拉召开的 Xtreme 游戏开发者大会,当时我和我的好多朋友站在一起,讨论 Premier Press 的游戏开发系列丛书。那个时候,我已经对编译器理论相关的话题进行了深入的研究——特别是如何将这些应用到游戏脚本中——也就是在那个时候我提出出版一本和脚本相关的书籍应该不错。恰好 André Lamothe 从旁边走过,“我们征求一下他的意见”我说,于是就把他拉了过来。“André, 想没想过在这些丛书中加入一本与游戏脚本相关的书籍?”我期望能得到“这个主意不错”或者“当然,已经有一部分了。”之类的答案。然而我得到的答案至少有一些让我惊讶。

“你为什么 not 写一本呢?”

这就是他所说的。除非你是在玩某种怪异版本的 Jeopardy!, 游戏规则就是你必须以书籍交易的形式来组织你的答案,而现在这确实是一个令我吃惊的答案。我眨着眼睛,考虑了很短的时间,并且马上说 OK。这也是我处理大多数重要决定所采取的方式,大多数事情当时都是无法预测的,这件事情就是一个很好的范例。不要质疑命运之神的存在。

关于这本书的渊源我们就不再详加论述了,我要感谢许多重要的人,他们在本书的写作过程中都给予了无私的帮助。如果不从 Mitzi Foster 开始的话就太说不过去了,Mitzi Foster 是这本书的 primer 编辑,她在这本书编写的过程中展示出了超人的忍耐力。在全部的章节完成并提交后需要进行大量的改写,这对一个编辑来说简直就是噩梦,她却是为数不多的能够忍受这一点的人之一,Mitzi Foster 以一贯友好和支持的态度从容冷静地完成了这些事情。

下来要感谢我的 copy 编辑 Kezia Endsley,你可能注意到了本书中几乎没有什么语法错误,甚至包括代码清单中的注释都非常清楚,这要感谢 Kezia Endsley。当然在最新的 Microsoft 编译器具注释语法检查功能, dancing monkey、robot dog, 或者 ethnically ambiguous baby 之前,这只是时间的问题,但是她的眼睛能够如此地明察秋毫,这一点不得不让人钦佩。

最后圆满解决《游戏脚本高级编程》的就是我的项目编辑 Estelle Manticas,在这个项目的后期她起到相当大的作用,当我们周围一团糟的时候,她仍然保持着某种幽默感。很少有人能够在压力很大的时候精心管理整本书的工作量,但是她使得这一切看起来很容易。

当然由于我年龄相对较小并喜欢像 NASA 那样大把大把地花钱,所以我还需要借助于其他人的帮助,毫不奇怪在这里我要把这些荣誉献给我的父母,我要感谢我的母亲把这本书的事情告诉给我的每个朋友、亲戚、老师以及我们家曾经认识的邮递员,另外,我还要感谢我的父亲,他担心在我工作的时候修理我房间外面的地板会影响到我的工作,因此他就决定每天凌晨左右才来干这些活。当然我不会忘记我的妹妹 Katherine,她经常坚持要我驾车送她上班,这也是唯一让我保持在适当时间醒来的事情。非常感谢他们!

最后我至少应该感谢 Lamothe 那个小伙子,确切地说我也许会被无尽的编码和手稿所

困扰，但 André 是这本书产生的真正原因（他也是这本书的技术编辑）。我要对他说“谢谢你让我经常袭击你的冰箱，教会我和电子工程相关的知识，给我带来的这么多免费的书本，不断回答我无尽的问题，可以用高尔夫俱乐部结束我们不断升温的讨论，当然还有其他各种无私的建议。大家都应该知道在工业界没有人会给予别人这么多的帮助，而我是为数不多的受益者之一”。

我还要特别感谢 John Romero，他从繁忙的日程中挤出时间来为本书作序。如果没有他，我可能得让我的母亲来完成这项工作了。

顺便提一下，正是因为我想快要出版这本书了，我再大声地说：感谢 Ironblayde，xms 和 Protoman，3 位有天赋的编程高手，以及定期与我在线交谈的少数几个人——他们听我不断激昂地演说，并鼓励我完成工作（如果不是因为他们的大力帮助，我可能就完不成这本书了）。

现在如果你们能够原谅我，我就结束这一部分了，我感觉就像是在签署一本年鉴。

关于作者

Alex Varanese 从 20 世纪 80 年代中期，也就是在他 5 岁时就被游戏开发给迷住了，他首先把眼光投向了——带着迷恋和某种不可解释的亲密——8 位的任天堂娱乐系统；同时他也对艺术有极大的热情，但是后来才成为一个真正的代码编写者，大约是在 15 岁左右，那个时候使用的是 QBASIC。18 岁的时候，他作为一个 Java 程序员在硅谷开始了他的程序员生涯，从事 J2EE 平台上的一些爆发的 B2B 项目，而在这之前大约一年他还是一个半自由撰稿人和室内图形设计者。

然而由于感觉到在办公室的生活太受限制，他又重新将注意力投向了游戏开发和对先进技术的追求。他现在是 eGameZone (<http://www.egamezone.net>) 的首席设计师和系统架构师，André LaMothe 的 Xtreme Games LLC 继任者。

他在空闲时间编程，渲染，以第三人称描述自己，并在弹出式广告上大肆宣传，在百老汇经典作品 *Dude, Where's My Car? The Musical* 中担任主角，同时表现出不可忽视的发展势头。

Alex Varanese 的 E-mail 是 alex@amvbooks.com，随时愿意并期望回答有关本书的任何问题，所以可以尽管向他提问！

来自系列丛书编辑的信

很久以前还在使用 8 位计算机的时候，你可能侥幸使用硬编码的方式表示你的逻辑、人工智能等。现在就像他们说的那样“忘记那些吧……”。游戏太复杂了，以至于不能只想到编码了——实际上，所有商业游戏的 99% 都是这样工作的：开发一个 3D 游戏引擎，接着在虚拟机的基础上使用脚本语言系统创建一个引擎接口（通常是一个高级语言）。脚本语言是供游戏开发者使用的，或者甚至是游戏的设计者，他们用脚本语言来创建游戏的真正逻辑以及整个游戏的行为。此外许多的标准编程规则，比如严格类型定义和单线程执行都在脚本语言里被打破了。实质上也就是将游戏开发的工作量划分开了，游戏设计者负责逻辑和游戏策划，游戏编程人员完成 3D 引擎、物理以及引擎的核心技术。

所以当决定在游戏中使用脚本的时候应该从哪里着手呢？当然在网络上有很多的资料，你可以试着在你的游戏中使用 Python、Lau 和其他的脚本语言，但是我认为你应该知道如何从头做起。这也就是游戏脚本精通大师所要完成的任务。这本书是一个怪物——Alex 介绍了你可以想象到的关于游戏脚本的每个细节。

这本书相对来说比较难，我们一直在讨论编译理论、虚拟机和多线程。然而 Alex 却从头开始，假设你对于脚本或者编译器一无所知，所以即使你是一个入门者，你也能够跟得上，前提是你要有能够有耐心地阅读每部分材料。在这本书结束的时候你就能够编写编译器和虚拟机了，能够用你的语言调用现有的 C/C++ 游戏引擎的接口——实质上说，你已经精通了游戏脚本！同时只要你活着就再也不会去编写另一个语法分析器了。

作为总结，如果你对游戏脚本感兴趣，并希望真正从中学到一些东西，那么这本书就是你所要的。而且这也是市场上关于这个话题的唯一一本。因为这是游戏脚本的领头羊，当我们自己策划这本书的时候，我们试图给你所需要的任何东西——并且我认为我们已经做到了，甚至做了更多。但是你才是最终的裁判！

真诚的，



丛书编辑

André LaMothe

关于本书配套光盘

本书所附光盘中存放了大量补充内容, 这些内容将有助于提高读者对本书的学习效果, 它们是按照如下所示的简单目录结构进行组织的:

- ❑ **Articles:** 存放了一些本书并未直接讨论的脚本编程方面的文章。
- ❑ **Programs:** 包含了本书各个章节中的所有代码和演示。该文件夹又会根据章节划分成各个子文件夹。例如, 第 12 章的代码和可执行文件就可以在目录 Programs/Chapter 12/中找到。在每个子文件内部, 你都可以找到一个 Read me! 文件, 它将会简单地介绍这些程序并提供如何编译这些程序的命令。
- ❑ **Software:** 一些我认为在某种程度上可以使用于脚本编程的软件, 例如 Flex 和 Bison、文本编辑器、十六进制编辑器, 以及语法分析器生成器。
- ❑ **Scripting Systems:** 该文件夹存放可以在游戏和程序中使用的脚本编程系统。
- ❑ **XtremeScript:** 在本书的最后, 我们设计出了 XtremeScript 脚本编程系统。我并不希望让读者通过掌握这些程序演示从而全面理解这个软件, 实际上, 我会将所有东西(编译器、汇编器、虚拟机以及单独的虚拟机控制台)都收集起来, 并将它们存放在一起。

每个文件夹都会存放有一个 Read Me!.txt 文件, 这个文件记录的有文件夹内容以及编译或者安装这些内容所需命令等重要信息。阅读这个文件的内容非常重要, 但是如果你发现你对一些问题仍然感到迷惑不解, 那么不要犹豫, 你就可以直接和我联系, 我的 E-mail 地址是 alex@amvbooks.com, 我将会时刻准备为你解答与本书相关的问题。

光盘界面

这个光盘中存放也是图形化的基于 HTML 的用户界面, 这样读者就可以更加方便地浏览该光盘中的内容了。不过, 因为它的用户界面是基于网页的, 因此需要一个 4.0 的浏览器。我建议读者使用微软的 IE 浏览器。

安装

安装过程其实相当的简单, 这些程序都有它们自己的可执行安装文件或者是自解压的安装文件, 而剩余的内容——即程序的演示和代码——也就会通过直接的从光盘拖放到硬盘从而实现“安装”。GUI 应该能够自动运行, 但是如果它不能自动运行的话, 那么读者就可以使用桌面上像 Windows IE 浏览器或者我的电脑等图标来查看这些内容, 当然, 也可以随心所欲的对你所需要的部分进行手工安装或者是拖动。

DirectX SDK

最后一点，读者需要使用 DirectX SDK 来浏览本书中的图形演示。如果你还没有安装这个软件，或者不是最新版本（编写本书的时候是 8.1），那么你就可以使用 CD-ROM GUI 来直接将它安装到你的系统上，不过，你也可以通过执行光盘中 DirectX/文件夹下的可执行安装文件来对其进行安装。

警告

从这个光盘中复制的文件都会带有 Archive 或者 Read-only 标记。因为光盘中的内容是不可以重写的，所以它们刚开始会设置这些标记；但是，如果你将它们拖放到硬盘上，这个限制也就不应该再存在了。不过，你的文件系统或者 shell 通常仍然会将这些标记保留下来，这样一来，你就必须要自己手工进行改变了。如果你忘记实现这种处理，那么程序演示源代码就只能是只读的了。在一台安装 Windows 操作系统的机器上，为了实现这种改变，你就需要选择所有你从光盘上复制过来的文件夹和/或文件，单击鼠标右键调出所有这些内容总体的属性对话框，然后去掉 Archive 和 Read-only 两个复选框上的标记。最后，单击“确定”按钮，你也就完成所有的处理了。

目 录

第 1 篇 脚本编程的基本原理

第 1 章 脚本编程概论.....	2
1.1 什么是脚本编程.....	2
1.2 结构化的游戏内容——一种简单的游戏设计方法.....	3
1.3 通过将逻辑和具体实现相互分离来改善这种方法.....	7
1.4 硬编码的危险性.....	9
1.5 将功能函数存放到外部文件.....	10
1.6 脚本是如何运行的.....	12
1.6.1 计算机程序设计概述.....	12
1.6.2 脚本编程概述.....	14
1.7 脚本系统的基本类型.....	16
1.7.1 面向过程的语言系统和面向对象的语言系统.....	16
1.7.2 基于命令的语言系统.....	17
1.7.3 动态链接的模块系统.....	18
1.7.4 编译型代码和解释型代码.....	19
1.7.5 现有的脚本设计方法.....	20
1.8 总结.....	21
第 2 章 脚本编程系统的应用.....	22
2.1 编写脚本的普遍目的.....	22
2.2 角色扮演游戏 (Role Playing Games, RPGs)	24
2.2.1 复杂而又有深度的故事情节.....	24
2.2.2 非玩家角色 (Non-Player Characters, NPCs)	25
2.2.3 物品和武器.....	31
2.2.4 敌人.....	34
2.3 第一人称射击游戏 (First-Person Shooters, FPSs)	38
2.3.1 物品、谜题和开关.....	39
2.3.2 敌人智能 (enemy AI)	44
2.4 总结.....	46

第 2 篇 基于命令的脚本

第 3 章 基于命令的脚本编程绪论	48
3.1 基于命令的脚本编程的基础知识	48
3.1.1 高级的引擎控制	49
3.1.2 命令	51
3.1.3 你这个领域中的大师	51
3.1.4 实际解决掉的一些问题	51
3.2 基于命令的脚本编程综述	52
3.2.1 游戏引擎的功能估计	52
3.2.2 载入和执行脚本	54
3.3 实现一种基于命令的语言	56
3.3.1 设计这种语言	56
3.3.2 编写脚本	56
3.3.3 实现	57
3.4 为一个游戏的介绍顺序编写脚本	67
3.4.1 语言	68
3.4.2 脚本	69
3.4.3 执行	69
3.5 为 RPG 游戏中某个角色的行为编写脚本	70
3.5.1 语言	71
3.5.2 改进语法	71
3.5.3 管理游戏中的角色	72
3.5.4 脚本	74
3.5.5 执行	75
3.5.6 游戏演示中的主循环	78
3.6 并发的脚本执行	80
3.7 总结	82
3.8 CD 上的内容	82
3.9 挑战	82
第 4 章 基于命令的高级脚本	84
4.1 新的数据类型	85
4.1.1 布尔变量	85
4.1.2 支持浮点数	85
4.1.3 通用目的符号常量	86

4.2 简单迭代和条件逻辑.....	92
4.2.1 条件逻辑和游戏标记.....	93
4.2.2 将代码按块分组.....	94
4.2.3 块列表.....	96
4.2.4 迭代逻辑.....	97
4.2.5 嵌套.....	98
4.3 基于事件的脚本.....	100
4.4 将脚本编译成二进制格式.....	102
4.4.1 提高执行速度.....	102
4.4.2 检测编译时的错误.....	103
4.4.3 恶意的脚本攻击.....	103
4.4.4 商用编译器是如何工作的.....	104
4.5 基本的脚本预处理.....	108
4.6 总结.....	111

第 3 篇 结构化的脚本编程语言概述

第 5 章 结构化的脚本编程系统介绍.....	114
5.1 总体的脚本编写结构.....	114
5.1.1 高级代码.....	115
5.1.2 低级代码.....	115
5.1.3 虚拟机.....	116
5.2 深入分析 XtremeScript	118
5.2.1 高级代码/编辑.....	119
5.2.2 低级代码/汇编.....	122
5.2.3 虚拟机.....	124
5.3 XtremeScript 系统	125
5.3.1 高级部分.....	125
5.3.2 低级部分.....	125
5.3.3 运行时间.....	125
5.4 总结.....	126
第 6 章 对现有脚本编程系统的集成和使用	127
6.1 集成 (Integration)	127
6.2 脚本系统的执行过程.....	131
6.3 跳动的外星人头像演示.....	132
6.4 Lua (以及基本的脚本编程概念)	135

6.4.1	Lua 系统一瞥.....	136
6.4.2	Lua 语言.....	137
6.4.3	将 Lua 语言与 C 语言相集成.....	153
6.4.4	高级的 Lua 话题.....	182
6.4.5	网络链接.....	183
6.5	Python.....	183
6.5.1	Python 系统一览.....	183
6.5.2	Python 语言.....	185
6.5.3	Python 与 C 的集成.....	200
6.5.4	高级话题.....	219
6.5.5	网络链接.....	220
6.6	Tcl.....	220
6.6.1	ActiveStateTcl.....	221
6.6.2	Tcl 语言.....	223
6.6.3	Tcl 和 C 的集成.....	241
6.6.4	高级主题.....	255
6.6.5	网络链接.....	255
6.7	你到底该使用哪个脚本编程系统呢.....	256
6.8	编写一个实际的游戏.....	258
6.9	总结.....	258
6.10	CD 上的内容.....	258
第 7 章	设计一个面向过程的脚本编程语言.....	259
7.1	语言的常见类型.....	259
7.1.1	汇编类型的语言 (Assembly-Style Languages).....	260
7.1.2	权衡利弊.....	263
7.2	函数.....	265
7.2.1	面向对象程序设计.....	268
7.2.2	XtremeScript 语言概述.....	269
7.2.3	语法和特性.....	271
7.2.4	保留字列表.....	280
7.3	总结.....	281

第 4 篇 设计和实现一种底层语言

第 8 章	汇编语言入门.....	284
8.1	什么是汇编语言.....	284
8.2	为什么现在讲述汇编.....	285

8.3 汇编语言是怎样工作的	285
8.3.1 指令	285
8.3.2 助记符和操作码	294
8.3.3 RISC 和 CISC	296
8.3.4 正交指令集	298
8.3.5 寄存器	298
8.3.6 堆栈	299
8.4 XVM 汇编简介	304
8.4.1 初始赋值	305
8.4.2 XVM 指令集	306
8.4.3 XASM 指示符	310
8.5 XVM 汇编语言概要	312
8.6 小结	313
第 9 章 构造 XASM 汇编器	314
9.1 一个简单的汇编器是如何运作的	315
9.1.1 指令的汇编	315
9.1.2 变量的汇编	317
9.1.3 操作数的汇编	319
9.1.4 字符串型字面量的汇编	321
9.1.5 跳转指令和函数调用的汇编	322
9.2 XASM 概述	326
9.2.1 内存管理	326
9.2.2 输入：一个 XVM 汇编脚本的结构	328
9.2.3 输出：一个 XVM 可执行文件的结构	338
9.3 汇编器的实现	345
9.3.1 基本的词法分析/语法分析理论	345
9.3.2 基本字符串处理	350
9.3.3 汇编器的框架	356
9.3.4 词法分析/属性字识别	376
9.3.5 错误处理	398
9.3.6 语法分析	400
9.3.7 构建 XSE 可执行文件	418
9.3.8 汇编过程	423
9.4 小结	427
9.5 CD 上的内容	428
9.6 挑战	428

第 5 篇 设计和实现虚拟机

第 10 章 汇编语言入门.....	430
10.1 虚拟机的灵魂.....	430
10.1.1 模拟硬件.....	430
10.1.2 虚拟机的主要组成部分.....	432
10.1.3 多线程.....	433
10.1.4 和主应用程序的集成.....	434
10.2 虚拟机生命周期的简单概述.....	434
10.2.1 装载脚本.....	435
10.2.2 从入口点开始执行.....	436
10.2.3 执行周期.....	436
10.2.4 函数调用.....	437
10.2.5 终止和关闭.....	439
10.3 XVM 原型的结构概述.....	440
10.3.1 脚本头.....	440
10.3.2 运行时值.....	441
10.3.3 指令流.....	442
10.3.4 运行时堆栈.....	443
10.3.5 函数表.....	444
10.3.6 主应用程序 API 调用表.....	445
10.3.7 最终的脚本结构.....	445
10.4 建立 XVM 原型.....	446
10.4.1 装载 XSE 可执行文件.....	446
10.4.2 结构接口.....	456
10.4.3 初始化虚拟机.....	471
10.4.4 执行周期.....	473
10.4.5 终止与关闭.....	488
10.5 概述.....	490
10.6 CD 上的内容.....	490
10.7 挑战.....	490
第 11 章 高级虚拟机的概念和问题.....	492
11.1 新一代虚拟机.....	492
11.2 多线程.....	492
11.2.1 多线程的基本原理.....	493

11.2.2 装载和存储多个脚本	503
11.2.3 执行多线程	511
11.3 主应用程序集成	515
11.3.1 和主应用程序并行运行脚本	515
11.3.2 集成接口简介	517
11.3.3 XVM 的公有接口 (Public Interface)	524
11.3.4 实现集成接口	526
11.3.5 增加线程优先级	550
11.4 演示最终的 XVM	557
11.5 小结	563
11.6 CD 上的内容	563
11.7 挑战	563

第 6 篇 编译高级代码

第 12 章 编译理论概论	566
12.1 编译理论概述	566
12.1.1 编译的几个阶段	567
12.1.2 XtremeScript 与 XASM 是如何工作的	578
12.1.3 高级编译理论	579
12.2 总结	587
第 13 章 词法分析器	588
13.1 基础知识	588
13.1.1 从字符到单词	589
13.1.2 属性标注	590
13.1.3 词法分析方法	590
13.2 词法分析器的工作	594
13.2.1 读入并保存文本文件	595
13.2.2 显示结果	596
13.2.3 错误处理	597
13.3 一个数值词法分析器	598
13.3.1 分词策略	598
13.3.2 完成这个演示程序	606
13.4 分析标识符和保留字	608
13.4.1 新的状态和属性符	609
13.4.2 测试文件	610