第一图书网, tushu007.com

# <<程序设计语言>>

## 图书基本信息

书名：<<程序设计语言>>

13位ISBN编号：9787121170676

10位ISBN编号：7121170671

出版时间：2012-7

出版时间：电子工业出版社

作者：Scott

页数：847

字数：1135000

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：http://www.tushu007.com

第一图书网, tushu007.com


## 前言

译　者　序　　Michael Scott的《程序设计语言——实践之路》（Programming Language Pragmatics）是一本优秀的教科书。
在网络时代计算技术飞速发展的背景下，各种创新的软硬件设计理念及实践如雨后春笋般层出不穷，如何将大量的信息组织起来，并突出其核心内容，是类似教材的作者们所面临的现实挑战。
而Scott这本教材最大的特点就在于，它紧紧抓住了处于计算机科学技术领域中心位置的主题——程序设计语言，为读者深入地探讨了程序设计语言及其实现的关键概念，同时将笔触延伸到编译技术、软件系统甚至软硬件体系结构等诸多领域。
　　本书系统地介绍了与程序设计语言相关的各种基本概念，内容涉及语言处理方面的具体细节，以及各种语言范式。
作者在讨论各个主题时，将对语言概念的描述与如何实现这些概念的具体说明从整体上结合在了一起，讲解清晰，并且给出了大量的实例。
这样，就使得读者通过学习本书，不仅能“知其然”——了解各种程序设计语言的实现中所做出的具体选择，还能够“知其所以然”——了解这些语言的设计者做出这些选择背后的逻辑和取舍。
　　本书的编排方式也非常灵活，各章的内容相对独立，每章都有大量随堂练习和课后习题，以帮助读者巩固学到的知识，并启发他们做进一步的思考。
随书附带的光盘上还包含了许多较深入的内容。
根据需要，这本教科书可以通过不同的方式组织起来，用于侧重点不同的教学课程，读者自学时也可以根据自己的情况自由选读。
　　作者在本书（第3版）中对内容做了大量的更新，最明显的就是增加了关于运行时程序管理的全新的第15章，以及关于并发的第12章的很多改写，增加了一些较新的主题。
这一版本更新了大量的实例（例如用X86上的C代码代替了Pascal），并且更多地采用了C#、Java 5、Python和Eiffel等现代语言的例子。
作者对第3版的改进还体现在许多细节方面，足以反映出作者之用心。
　　本书在翻译的过程中，得到了博文视点总经理郭立老师的支持与指导，她对IT专业书籍翻译的质量非常重视，使我们对本书的翻译不敢有丝毫的懈急，力求达到精益求精。
感谢本书的策划编辑刘皎，是她持续的鼓励给予我们信心让我们继续下去。
还要感谢那些为本书的翻译做出贡献的所有人，他们都是默默无闻的后台工作者。
　　诚然，限于译者水平，书中难免含有一些错误和理解不到位的地方，敬请读者朋友批评指正。
　　译者　　2011年中秋节于北京

第一图书网, tushu007.com

# <<程序设计语言>>

## 内容概要

这是一本很有特色的计算机教材，其核心是讨论程序设计语言的基本原理和技术。
本书融合了传统的程序设计语言教科书和编译教科书的有关知识，并增加了一些有关汇编层体系结构的材料，以满足没学过计算机组织的学生们的需要。
书中通过各种语言的例子，阐释了程序设计语言的重要基础概念，讨论了各种概念之间的关系，解释了语言中许多结构的形成和发展过程，以及它们演化为今天这种形式的根源。
书中还详细讨论了编译器的工作方式和工作过程，说明它们对源程序做了什么，以及为什么要那样做。
书的每章最后附有复习题和一些更具挑战性的练习与探索。
这些练习的特别价值在于引导学生进一步深入理解各种语言和技术。
本书第3版新增了关于运行时程序管理的讨论，对关于并发的一章做了重大的改写，并更新了大量的实例。

这本教材在美国大学已使用了二十余年，目前被欧美许多重要大学用于“程序设计语言”或者“软件系统”课程。

第一图书网, tushu007.com



## 作者简介

作者:(美)Scott

第一图书网, tushu007.com

# <<程序设计语言>>

## 书籍目录

第1部分 基础
第1章 引言
1.1 语言设计的艺术
1.2 程序设计语言的谱系
1.3 为什么要研究程序设计语言？

1.4 编译和解释
1.5 程序设计环境
1.6 编译概览
1.6.1 词法和语法分析
1.6.2 语义分析和中间代码生成
1.6.3 目标代码生成
1.6.4 代码改进
1.7 总结和注记
1.8 练习
1.9 探索
1.10 有关参考文献
第2章 程序设计语言的语法
2.1 描述语法：正则表达式和上下文无关文法
2.1.1 单词和正则表达式
2.1.2 上下文无关文法
2.1.3 推导和语法分析树
2.2 扫描
2.2.1 生成一个有穷自动机
2.2.2 扫描器代码
2.2.3 表格驱动的扫描
2.2.4 词法错误
2.2.5 编译指示
2.3 语法分析
2.3.1 递归下降
2.3.2 表格驱动的自上而下语法分析
2.3.3 自下而上的语法分析
2.3.4 语法错误
2.4 理论基础
2.4.1 有穷自动机 13
2.4.2 下推自动机 18
2.4.3 文法和语言类 19
2.5 总结和注记
2.6 练习
2.7 探索
2.8 有关参考文献
第3章 名字、作用域和约束
3.1 约束时间的概念
3.2 对象生存期和存储管理
3.2.1 静态分配

第一图书网, tushu007.com


3.2.2 基于栈的分配
3.2.3 基于堆的分配
3.2.4 废料收集
3.3 作用域规则
3.3.1 静态作用域
3.3.2 嵌套子程序
3.3.3 声明的顺序
3.3.4 模块
3.3.5 模块类型和类
3.3.6 动态作用域
3.4 作用域的实现
3.4.1 符号表
3.4.2 关联表和中心引用表
3.5 作用域中名字的含义
3.5.1 别名
3.5.2 重载
3.5.3 多态性及相关概念
3.6 引用环境的约束
3.6.1 子程序闭包
3.6.2 一级值和非受限生存期
3.6.3 对象闭包
3.7 宏扩展
3.8 分别编译
3.8.1 C的分别编译
3.8.2 包和自动头文件推理
3.8.3 模块分层结构
3.9 总结和注记
3.10 练习
3.11 探索
3.12 有关参考文献
第4章 语义分析
4.1 语义分析器所扮演的角色
4.2 属性文法
4.3 属性求值
4.4 动作例程
4.5 属性的空间管理
4.5.1 自下而上求值
4.5.2 自上而下求值
4.6 语法树的标注
4.7 总结和注记
4.8 练习
4.9 探索
4.10 有关参考文献
第5章 目标机体系结构
5.1 存储器层次结构
5.2 数据表示
5.2.1 整数算术

第一图书网, tushu007.com


5.2.2 浮点数算术
5.3 指令集体系结构
5.3.1 寻址模式
5.3.2 条件和分支
5.4 体系结构和实现
5.4.1 微程序设计
5.4.2 微处理器
5.4.3 RISC
5.4.4 多线程和多核
5.4.5 两个示例体系结构：x86和MIPS
5.5 为新型处理器做编译
5.5.1 保持流水线满
5.5.2 寄存器分配
5.6 总结和注记
5.7 练习
5.8 探索
5.9 有关参考文献
第2部分 语言设计的核心问题
第6章 控制流
6.1 表达式求值
6.1.1 优先级和结合性
6.1.2 赋值
6.1.3 初始化
6.1.4 表达式中的顺序问题
6.1.5 短路求值
6.2 结构化和非结构化的流程
6.2.1 goto的结构化替代品
6.2.2 继续
6.3 顺序执行
6.4 选择
6.4.1 短路条件
6.4.2 Case/Switch语句
6.5 迭代
6.5.1 枚举控制的循环
6.5.2 组合循环
6.5.3 迭代器
6.5.4 Icon的生成器
6.5.5 逻辑控制的循环
6.6 递归
6.6.1 迭代和递归
6.6.2 应用序和正则序求值
6.7 非确定性
6.8 总结和注记
6.9 练习
6.10 探索
6.11 有关参考文献
第7章 数据类型

第一图书网, tushu007.com


7.1 类型系统
7.1.1 类型检查
7.1.2 多态性
7.1.3 “ 类型 ”的含义
7.1.4 类型的分类
7.1.5 正交性
7.2 类型检查
7.2.1 类型等价
7.2.2 类型相容性
7.2.3 类型推理
7.2.4 ML类型系统
7.3 记录（结构）与变体（联合）
7.3.1 语法和运算
7.3.2 存储布局及其影响
7.33 with语句
7.3.4 变体记录（联合）
7.4 数组
7.4.1 语法和操作
7.4.2 维数、上下界和分配
7.4.3 内存布局
7.5 字符串
7.6 集合
7.7 指针和递归类型
7.7.1 语法和操作
7.7.2 悬空引用
7.7.3 废料收集
7.8 表
7.9 文件和输入/输出
7.9.1 交互式I/O
7.9.2 基于文件的I/O
7.9.3 正文I/O
7.10 相等检测和赋值
7.11 总结和注记
7.12 练习
7.13 探索
7.14 有关参考文献
第8章 子程序和控制抽象
8.1 回顾栈的布局
8.2 调用序列
8.2.1 区头向量
8.2.2 案例研究：在MIPS上实现C，在x86上实现Pascal
8.2.3 寄存器窗口
8.2.4 内联展开
8.3 参数传递
8.3.1 参数模式
8.3.2 名字调用
8.3.3 特殊目的的参数

第一图书网, tushu007.com


8.3.4 函数返回
8.4 泛型子程序和模块
8.4.1 不同的实现方法
8.4.2 泛型参数的约束条件
8.4.3 隐式实例化
8.4.4 C++、Java和C#中的泛型
8.5 异常处理
8.5.1 异常的定义
8.5.2 异常的传播
8.5.3 异常的实现
8.6 协作程序
8.6.1 栈分配
8.6.2 转移
8.6.3 迭代器的实现
8.6.4 离散事件模拟
8.7 事件
8.7.1 顺序处理程序
8.7.2 基于线程的处理程序
4.8 总结和注记
8.9 练习
8.10 探索
8.11 有关参考文献
第9章 数据抽象和面向对象
9.1 面向对象程序设计
9.2 封装和继承
9.2.1 模块
9.2.2 类
9.2.3 嵌套（内层类）
9.2.4 类型扩展
9.2.5 不使用继承扩展
9.3 初始化和终结处理
9.3.1 构造函数的选择
9.3.2 引用和值
9.3.3 执行顺序
9.3.4 废料收集
9.4 动态方法约束
9.4.1 虚方法和非虚方法
9.4.2 抽象类
9.4.3 成员查找
9.4.4 多态性
9.4.5 对象闭包
9.5 多重继承
9.5.1 语义歧义性
9.5.2 复本式继承
9.5.3 共享继承
9.5.4 混入式继承
9.6 重温面向对象的程序设计

第一图书网, tushu007.com


9.6.1 Smalltalk的对象模型
9.7 总结和注记
9.8 练习
9.9 探索
9.10 有关参考文献
第3部分 其他程序设计模型
第10章 函数式语言
10.1 历史渊源
10.2 函数式程序设计的概念
10.3 Scheme回顾/简介
10.3.1 约束
10.3.2 表和数
10.3.3 相等检测和检索
10.3.4 控制流和赋值
10.3.5 程序作为表
10.3.6 一个扩展的实例：DFA模拟
10.4 重温求值顺序
10.4.1 严格求值和惰性求值
10.4.2 I/O：流和单体
10.5 高阶函数
10.6 理论基础
10.6.1 lambda 演算
10.6.2 控制流
10.6.3 结构
10.7 函数式程序设计展望
10.8 总结和注记
10.9 练习
10.10 探索
10.11 有关参考文献
第11章 逻辑式语言
11.1 逻辑式程序设计的概念
11.2 Prolog
11.2.1 归结和合一
11.2.2 表
11.2.3 算术
11.2.4 搜索/执行顺序
11.2.5 一个较大的实例：九宫棋
11.2.6 命令式控制流
11.3 理论基础
11.3.1 子句形式
11.3.2 局限性
11.3.3 Skolem
11.4 逻辑式程序设计的展望
11.4.1 没有覆盖的逻辑部分
11.4.2 执行顺序
11.4.3 否定和“闭世界”假设
11.5 总结和注记

第一图书网, tushu007.com


11.6 练习
11.7 探索
11.8 有关参考文献
第12章 并发
12.1 基础和动力
12.1.1 多线程程序的各种情况
12.1.2 多处理器体系结构
12.2 并发程序设计基础
12.2.1 通信和同步
12.2.2 语言和库
12.2.3 创建线程的语法
12.2.4 线程的实现
12.3 实现
12.3.1 忙等待同步
12.3.2 非阻塞算法
12.3.3 内存一致模型
12.3.4 调度器的实现
12.3.5 信号量
12.4 语言级机制
12.4.1 管程
12.4.2 条件临界区域
12.4.3 Java中的同步
12.4.4 事务存储
12.4.5 隐式同步
12.5 消息传递
12.5.1 通信对方的命名
12.5.2 发送
12.5.3 接收
12.5.4 远程过程调用
12.6 总结和注记
12.7 练习
12.8 探索
12.9 有关参考文献
第13章 脚本语言
13.1 什么是脚本语言？

13.1.1 公共特性
13.2 问题领域
13.2.1 外壳（命令）语言
13.2.2 文字处理和报表生成
13.2.3 数学和统计
13.2.4 “粘结”语言和通用脚本
13.2.5 扩充语言
13.3 万维网脚本
13.3.1 CGI脚本
13.3.2 嵌入式服务器端脚本
13.3.3 客户端脚本

第一图书网, tushu007.com


13.3.4 Java小程序
13.3.5 XSLT
13.4 新特征
13.4.1 名字和作用域
13.4.2 串和模式匹配
13.4.3 数据类型
13.4.4 面向对象
13.5 总结和注记
13.6 练习
13.7 探索
13.8 有关参考文献
第4部分 对实现的近距离考查
第14章 构造可运行的程序
14.1 后端编译器结构
14.1.1 一种可行的多阶段组织
14.1.2 阶段和遍
14.2 中间形式
14.2.1 Diana
14.2.2 gcc中间形式
14.2.3 基于栈的中间形式
14.3 代码生成
14.3.1 一个属性文法实例
14.3.2 寄存器分配
14.4 地址空间组织
14.5 汇编
14.5.1 指令发射
14.5.2 为名字指定地址
14.6 连接
14.6.1　重定位和名字解析
14.6.2 类型检查
14.7 动态连接
14.7.1 与定位无关的代码
14.7.2 完全动态连接（惰性连接）
14.8 总结和注记
14.9 练习
14.10 探索
14.11 有关参考文献
第15章 运行时程序管理
15.1 虚拟机
15.1.1 Java虚拟机
15.1.2 公共语言基础架构
15.2 机器码的迟绑定
15.2.1 即时和动态编译
15.2.2 二进制翻译
15.2.3 二进制重写
15.2.4 移动代码和沙箱
15.3 审查/自反

第一图书网, tushu007.com


15.3.1 自反
15.3.2 符号调试
15.3.3 性能分析
15.4 总结和注记
15.5 练习
15.6 探索
15.7 有关参考文献
第16章 代码改进
16.1 代码改进的阶段
16.2 窥孔优化
16.3 基本块内的冗余删除
16.3.1 一直使用的实例
16.3.2 值编号
16.4 全局冗余删除和数据流分析
16.4.1 SSA（静态单赋值）形式和全局值编号
16.4.2 全局公共子表达式删除
16.5 循环改进I
16.5.1 循环不变量
16.5.2 归纳变量
16.6 指令调度
16.7 循环改进II
16.7.1 循环展开和软件流水线
16.7.2 循环重排
16.8 寄存器分配
16.9 总结和注记
16.10 练习
16.11 探索
16.12 有关参考文献
附录A 本书中提到的程序设计语言
附录B 语言设计和语言实现
附录C 编号示表

第一图书网, tushu007.com


## 章节摘录

版权页： 插图： 超级计算机 超级计算虽然与计算机行业的其他领域相比在财务上相形见绌，但是它在计算机技术的发展以及人类知识的进步方面，始终扮演着远超其实际投入的角色。
超级计算机随着时间的推移已经发生了巨大的变化，并且仍然继续以非常快的速度发展着。
不过，它们始终都是并行的机器。
由于缓存一致性的复杂性，很难构造出大型的共享存储机器。
SGI销售拥有512个处理器（1024个核）的机器。
Cray构造出了更大的共享存储机器，但是并不能缓存远程的位置。
然而，传统的向量机器在很大程度上不仅被大型多处理器机器替代，而且也被相当数量的较小多处理器或非常多的连接到高性能网络上的商用（主流）单处理器机器替代。
随着网络技术“向下流淌”到更广泛的市场中，这些机器又被由商用处理器（通常是多核的）和商用网络（吉比特以太网或无限带宽）组成的集群取代。
截止到2008年，集群已经在从大量的服务器场到除极快的超级计算机站点之外的所有领域中都处于优势地位。
Google、Amazon或eBay等大规模在线服务通常都是由成百甚至上千个处理器支持的（以Google为例，这一数字也许达到数十万）。
如今最快的机器是由特殊的高密度、低功能多核芯片构成的。
IBM的BlueGene／P系统使用了4核、16瓦PowerPC处理器。
IBM最近完成的RoadRunner系统（2008年6月时世界上最快的系统）使用的是90瓦PowerXCell处理器，与Sony Playstation 3中使用的处理器类似。
每个Cell处理器都包含一个双线程PowerPC和8个小向量核。
根据当前的趋势，看起来未来的高端和商用机器，都将变得越来越密集和多样化。
从程序设计语言的角度来看，超级计算的特殊挑战在于适应不统一的访问时间和（在大多数情况下）缺少硬件对跨整个机器共享访问的支持。
如今的超级计算机在编程时，主要是使用消息传递库（特别是MPI）和局部及远程存储访问之间存在显著区别的语言和库来进行的。

第一图书网, tushu007.com

# <<程序设计语言>>

## 编辑推荐

《程序设计语言:实践之路(第3版)》在美国大学已使用了二十余年，目前被欧美许多重要大学用于“程序设计语言”或者“软件系统”课程。
《程序设计语言:实践之路(第3版)》适合高年级本科生或者一年级研究生使用，许多内容对专业程序员也很有价值。

第一图书网, tushu007.com


## 名人推荐

对于语言设计和实现而言，本书是一本出色的入门书。
它不仅阐述了我们使用的那些语言背后的理论基础，还阐明了计算机体系结构的发展是如何引导这些理论的，以及这些理论将如何继续发展，以面对利用多核硬件的挑战。
——Tim Harris，微软研究院

第一图书网, tushu007.com



## 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:http://www.tushu007.com