

PEARSON

原书第7版

汇编语言 基于x86处理器

[美] 基普·欧文 (Kip Irvine) 著 贺莲 龚奕利 译
佛罗里达国际大学 武汉大学

Assembly Language for x86 Processors
Seventh Edition

Assembly
Language

FOR x86 PROCESSORS
Seventh Edition

Kip Irvine



机械工业出版社
China Machine Press

计 算 机 科 学 丛 书

原书第7版

汇编语言

基于x86处理器

[美] 基普·欧文 (Kip Irvine) 著 贺莲 龚奕利 译
佛罗里达国际大学 武汉大学

Assembly Language for x86 Processors
Seventh Edition

*Assembly
Language*

FOR x86 PROCESSORS
Seventh Edition

Kip Irvine



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

汇编语言：基于 x86 处理器（原书第 7 版）/（美）欧文（Irvine, K.）著；贺莲，龚奕利译．—北京：机械工业出版社，2016.3

（计算机科学丛书）

书名原文：Assembly Language for x86 Processors, Seventh Edition

ISBN 978-7-111-53036-7

I. 汇… II. ①欧… ②贺… ③龚… III. 汇编语言—程序设计 IV. TP313

中国版本图书馆 CIP 数据核字（2016）第 037015 号

本书版权登记号：图字：01-2015-7581

Authorized translation from the English language edition, entitled *Assembly Language for x86 Processors, Seventh Edition* (ISBN: 978-0-13-376940-1) by Kip Irvine, published by Pearson Education, Inc., Copyright © 2015, 2011, 2007, 2003.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Chinese simplified language edition published by Pearson Education Asia Ltd., and China Machine Press Copyright © 2016.

本书中文简体字版由 Pearson Education（培生教育出版集团）授权机械工业出版社在中华人民共和国境内（不包括中国台湾地区和香港、澳门特别行政区）独家出版发行。未经出版者书面许可，不得以任何方式抄袭、复制或节录本书中的任何部分。

本书封底贴有 Pearson Education（培生教育出版集团）激光防伪标签，无标签者不得销售。

本书是汇编语言课程的经典教材，系统介绍 x86 和 Intel64 处理器的汇编语言编程与架构。前 9 章为汇编语言的核心概念，包括：汇编语言基础，x86 处理器架构，数据传送、寻址和算术运算，过程，条件处理，整数运算，高级过程，以及字符串和数组。后 4 章介绍结构和宏、MS-Windows 编程、浮点数处理和指令编码，以及高级语言接口。

本书内容翔实，案例丰富，极具逻辑性和系统性，不仅可作为汇编语言课程的教材，还可作为计算机系统基础和体系结构基础方面的教材。

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码 100037）

责任编辑：和 静

责任校对：董纪丽

印 刷：中国电影出版社印刷厂

版 次：2016 年 4 月第 1 版第 1 次印刷

开 本：185mm×260mm 1/16

印 张：35.5

书 号：ISBN 978-7-111-53036-7

定 价：99.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：(010) 88378991 88361066

投稿热线：(010) 88379604

购书热线：(010) 68326294 88379649 68995259

读者信箱：hzsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问：北京大成律师事务所 韩光 / 邹晓东

文艺复兴以来，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的优势，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅肇划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自1998年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与Pearson, McGraw-Hill, Elsevier, MIT, John Wiley & Sons, Cengage等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出Andrew S. Tanenbaum, Bjarne Stroustrup, Brain W. Kernighan, Dennis Ritchie, Jim Gray, Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, Abraham Silberschatz, William Stallings, Donald E. Knuth, John L. Hennessy, Larry L. Peterson等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力相助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专门为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方式如下：

华章网站：www.hzbook.com

电子邮件：hzjsj@hzbook.com

联系电话：(010) 88379604

联系地址：北京市西城区百万庄南街1号

邮政编码：100037



译者序

Assembly Language for x86 Processors, Seventh Edition

大多数计算机专业的学生并不太喜欢也不太愿意学习汇编语言，因为它和机器硬件结合得非常紧密，使用起来不怎么得心应手，在解决问题方面，好像也不如其他高级语言有用。在教学过程中，不止一次听到学生抱怨说：“现在都不用汇编语言了，为什么还要学它？”彼时，受限于时间和授课内容，总觉得不能完全讲清楚这种语言的有效性和重要性。因此，当看到这本书后，我们就想要将它介绍给大家。

本书译自《Assembly Language for x86 Processors》第7版，内容包括：基本概念，x86处理器架构，汇编语言基础，数据传输、寻址和算术运算，过程，条件处理，整数运算，高级过程，字符串和数组，结构和宏，MS-Windows编程，浮点数处理和指令编码，高级语言接口等。本书还包括64位CPU架构和编程的内容，以及64位的子程序库Irvine64。

如果读者想了解BIOS编程、MS-DOS服务等内容，可以登录英文书配套网站进行阅读。同时，网站上还提供了VideoNotes教学视频直观演示汇编语言的基本概念。

本书具有丰富的习题，并按照不同要求与难度分为简答题、算法基础练习以及编程练习。因此，做习题的过程就是一个循序渐进、学以致用过程。相信完成这些习题后，大家就不再会觉得汇编语言有多么遥远了。

在此感谢机械工业出版社华章公司的朱劼编辑，感谢她向我们推荐了这本书，以及在翻译过程中给予我们的支持和帮助。

虽然在翻译过程中我们尽量做到认真细致，对每一个有疑问的点都进行了讨论，但是由于能力所限，还是会存在错误与疏漏，希望广大读者批评指正，同时我们也会将勘误更新到网站。

贺莲 龚奕利

2015年9月于珞珈山

本书介绍 x86 和 Intel64 处理器的汇编语言编程与架构，适合作为下述几类大学课程的教材：

- 汇编语言编程
- 计算机系统基础
- 计算机体系结构基础

学生使用 Intel 或 AMD 处理器，用 Microsoft 宏汇编器（Microsoft Macro Assembler, MASM）编程，MASM 运行在 Microsoft Windows 最新的版本上。尽管本书的初衷是作为大学生的编程教材，但它也是计算机体系结构课程的有效补充。本书广受欢迎，前几个版本已被翻译为多种语言。

重点主题 本版所含主题可以自然过渡到讲述计算机体系结构、操作系统和编写编译器的后续课程：

- 虚拟机概念
- 指令集架构
- 基本布尔运算
- 指令执行周期
- 内存访问和握手
- 中断和轮询
- 基于硬件的 I/O
- 浮点数二进制表示

其他主题则专门针对 x86 和 Intel64 架构：

- 受保护的内存和分页
- 实地址模式的内存分段
- 16 位中断处理
- MS-DOS 和 BIOS 系统调用（中断）
- 浮点单元架构和编程
- 指令编码

本书中的某些例子还可以用于计算机科学课程体系中的后续课程：

- 搜索与排序算法
- 高级语言结构
- 有限状态机
- 代码优化示例

第 7 版的新内容

这一版增加了对程序示例的讨论，添加了更多的复习题和关键术语，介绍了 64 位编程，降低了对子程序库的依赖性。具体内容如下：

- 本版前面的几章现在包含了以 64 位 CPU 架构和编程为主的小节，并且还创建了子程序库的 64 位版本 Irvine64。
- 修改、替换了很多复习题和练习，部分题目从章节内移动到该章末尾，且习题分为两部分：简答题和算法基础练习。后者要求学生编写一小段代码实现一个目标。
- 每章有一节为关键术语，列出了新的术语和概念，以及新的 MASM 伪指令和 Intel 指令。
- 添加了新的编程练习，删除了一些旧习题，并对一些现有的练习进行了修改。
- 本书对作者子程序库的依赖性大大减低。鼓励学生自己调用系统函数，并使用 Visual Studio 调试器单步执行程序。Irvine32 和 Irvine64 链接库可以帮助学生处理输入 / 输出，但是不强制要求使用它们。
- 作者录制的新视频教程涵盖了本书的基本内容，并已添加到 Pearson 网站[Ⓔ]。

本书仍然关注其首要目标，即教授学生编写并调试机器级程序。它不能代替计算机体系结构的完整教材，但它确实在告诉学生计算机工作原理的基础上，给出了编写软件的第一手经验。我们认为，理论联系实际能让学生更好地掌握知识。在工程课程中，学生构建原型；在计算机体系结构课程中，学生应编写机器级程序。在这些课程里，学生都能获得难忘的经验，从而有信心在任何 OS/ 面向机器的环境中工作。

保护模式编程是纸版章节（第 1 章~第 13 章）重中之重。因此，学生需要在最新版本 Microsoft Windows 环境下创建并运行 32 位和 64 位程序。其他 4 章是电子版[Ⓔ]，讲述 16 位编程。这些章包含了 BIOS 编程、MS-DOS 服务、键盘和鼠标输入、视频编程和图形图像内容。其中一章为磁盘存储基础，还有一章为高级 DOS 编程技术。

子程序库 本书为学生提供了三个版本的子程序库，用于基本输入 / 输出、模拟、计时和其他有用的任务。Irvine32 和 Irvine64 链接库运行于保护模式。16 位版本的链接库 (Irvine16.lib) 运行于实地址模式，且只用于第 14 章~第 17 章[Ⓔ]。这些库的完整源代码见于配套的网站。链接库是为了使用方便，而不是为了阻止学生学习如何自行对输入 - 输出编程。鼓励学生创建自己的链接库。

所含软件与示例 所有示例程序均在 Microsoft Visual Studio 2012 下，用 Microsoft Macro Assembler Version 11.0 进行了验证。此外，还提供了批处理文件允许学生用 Windows 命令行汇编和运行应用程序。第 14 章中的 32 位 C++ 应用程序已用 Microsoft Visual C++ .NET 测试。本书的内容更新与勘误参见配套的网站，其中包括了一些额外的编程项目，老师可以在章节结束的时候布置给学生。

总体目标

本书的以下目标旨在提高学生对汇编语言相关知识的兴趣并拓展知识面：

- Intel 和 AMD 处理器架构与编程；
- 实地址模式和保护模式编程；
- 汇编语言伪指令、宏、运算符与程序结构；

Ⓔ、Ⓕ、Ⓖ 这些内容属于付费内容，需要的读者可向培生教育集团北京代表处购买，电话：010-57355169/57355171，电子邮件：service.cn@pearson.com。——编辑注

- 编程方法，展示了如何用汇编语言创建系统级软件工具和应用程序；
- 计算机硬件操作；
- 汇编语言程序、操作系统和其他应用程序之间的交互作用。

本书的目标之一是帮助学生以机器级的思维方式来处理编程问题。将 CPU 视为交互工具，学习尽可能直接地监控其操作是很重要的。调试器是程序员最好的朋友，不仅可以捕捉错误，还可以用作学习 CPU 和操作系统的教学工具。我们鼓励学生探查高级语言的内部机制，并能意识到大多数编程语言都被设计为可移植的，因此，也独立于其运行的主机。除了短小的示例外，本书还有几百个可运行的程序来演示书中讲述的指令和思想。本书结尾有参考资料，包括 MS-DOS 中断和指令助记符指南。

背景知识 读者应至少能熟练使用一种高级语言进行编程，比如 Python、Java、C 或 C++。本书有一章涉及 C++ 接口，因此，如果手边有编译器将会非常有帮助。本书不仅已经用于计算机科学和管理信息系统专业课堂，而且还用于其他工程课程。

特点

完整的程序清单 配套的网站包含了补充资料、学习指南，以及本书全部示例的源代码。本书还提供了丰富的链接库，其中包括 30 多个过程，可以简化用户输入-输出、数字处理、磁盘和文件处理，以及字符串处理。课程初期，学生可以用这个链接库来改进自己编写的程序。之后，学生可以自行编写过程并将它们添加到链接库中。

编程逻辑 本书用两章的篇幅重点介绍了布尔逻辑和位操作，并且有意识地尝试将高级编程逻辑与底层机器细节对应起来。这有助于学生创建更有效的实现，且有助于他们更好地理解编译器是如何生成目标代码的。

硬件和操作系统概念 本书前两章介绍基础硬件和数据表示概念，包括二进制数、CPU 架构、状态标志和内存映射。概述硬件和以历史的角度审视 Intel 处理器系列可以帮助学生更好地理解其目标计算机系统。

结构化程序设计方法 从第 5 章开始，关注重点为过程和功能分解。同时，提供了更复杂的编程练习，要求学生在编码之前把设计作为重点。

Java 字节码和 Java 虚拟机 第 8 章和第 9 章解释了 Java 字节码的基本操作，并给出了简短的演示例子。很多短示例不仅给出了反汇编字节码形式，还给出了详细的步骤解释。

磁盘存储概念 学生从硬件和软件的角度学习基于 MS-Windows 的磁盘存储系统的基本原理。

创建链接库 学生不仅可以自由地把自己编写的过程添加到本书链接库，还可以创建新的链接库。他们要学习用工具箱方法进行编程，并编写多个程序可以共用的代码。

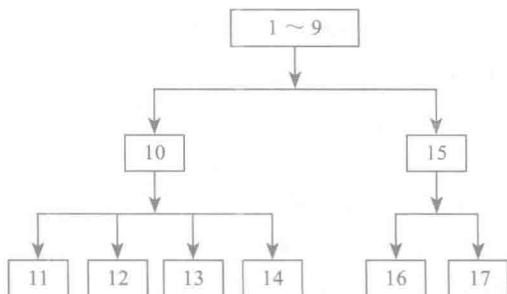
宏和结构 本书用一章专门描述创建结构、联合以及宏，这些对汇编语言编程和系统编程是非常重要的。条件宏和高级运算符使得宏更加专业。

高级语言接口 本书用一章专门描述汇编语言与 C 和 C++ 的接口。对于想要从事高级语言编程工作的学生而言，这是一项重要的工作技能。他们可以学习代码优化，还可以通过例子了解 C++ 编译器是如何优化代码的。

教学辅助 所有的程序清单都在网上。同时向教师提供了测试库、复习题答案、编程练习的解决方案，以及每章的 PPT。

章节说明

第 1 章~第 9 章为汇编语言核心概念,需要按顺序学习。后面的章节则可以自由选择。下面的章节示意图展示了后续章节与其他章节知识之间的依赖关系。



第 1 章 基本概念: 汇编语言的应用、基本概念、机器语言和数据表示。

第 2 章 x86 处理器架构: 基本微计算机设计、指令执行周期、x86 处理器架构、Intel64 架构、x86 内存管理、微计算机组件、输入-输出系统。

第 3 章 汇编语言基础: 介绍汇编语言、链接和调试、常量和变量定义。

第 4 章 数据传送、寻址和算术运算: 简单的数据传送和算术运算指令、汇编-链接-执行周期、运算符、伪指令、表达式、JMP 和 LOOP 指令、间接寻址。

第 5 章 过程: 与外部链接库的链接、描述本书链接库、堆栈操作、过程的定义和使用、流程图、自顶向下的结构设计。

第 6 章 条件处理: 布尔和比较指令、条件跳转和循环、高级逻辑结构、有限状态机。

第 7 章 整数运算: 移位和循环移位指令及其应用、乘法和除法、扩展加法和减法、ASCII 和压缩十进制运算。

第 8 章 高级过程: 堆栈参数、局部变量、高级 PROC 和 INVOKE 伪指令、递归。

第 9 章 字符串和数组: 字符串原语、操作字符和整数数组、二维数组、排序和检索。

第 10 章 结构和宏: 结构、宏、条件汇编伪指令、定义重复块。

第 11 章 MS-Windows 编程: 保护模式内存管理概念、用 Microsoft-Windows API 显示文本和颜色, 动态内存分配。

第 12 章 浮点数处理与指令编码: 浮点数二进制表示和浮点运算。学习 IA-32 浮点单元编程。理解 IA-32 机器指令编码。

第 13 章 高级语言接口: 参数传递规范、内嵌汇编代码、将汇编语言模块链接到 C 和 C++ 程序。

附录 A MASM 参考知识

附录 B x86 指令集

附录 C “本节回顾” 问题答案

下面的章节和附录由配套网站提供[⊖]:

第 14 章 16 位 MS-DOS 编程: 内存组织、中断、函数调用、标准 MS-DOS 文件 I/O 服务。

第 15 章 磁盘基础知识: 磁盘存储系统、扇区、簇、目录、文件分配表、处理 MS-

⊖ 这些内容属于付费内容, 需要的读者可向培生教育出版集团北京代表处购买, 电话: 010-57355169/57355171, 电子邮件: service.cn@pearson.com。——编辑注

DOS 错误码、驱动器和目录操作。

第 16 章 BIOS 编程：键盘输入、视频文本、图形、鼠标编程。

第 17 章 高级 MS-DOS 编程：自定义设计段、运行时程序结构、中断处理、用 I/O 端口的硬件控制。

附录 D BIOS 和 MS-DOS 中断

附录 E “本节回顾” 问题答案 (第 14 章~第 17 章)

教师和学生资源

教师资源^①

下面受保护的教师资源见配套网站 www.pearsonhighered.com/irvine：

- PPT 讲义
- 教师解题手册

学生资源

学生通过位于 www.pearsonhighered.com/irvine 的出版社网站可以找到本书作者的网站链接。下述资源位于 www.asmirvine.com，且不需要用访问卡：

- Getting Started (入门)，循序渐进的完整教程，帮助学生设置 Visual Studio 进行汇编语言编程。
- 与汇编语言编程主题相关的补充读物。
- 本书全部示例程序的完整代码，以及作者补充链接库的源代码。
- Assembly Language Workbook (汇编语言工作手册)，一个交互式的工作手册，其中包括数值转换、寻址模式、寄存器使用、调试编程和浮点二进制数。内容页面是可以自定义的 HTML 文档，帮助文件为 Windows 帮助格式。
- 调试工具：Microsoft Visual Studio 调试器用法教程。

致谢

非常感谢培生教育 (Pearson Education) 计算机科学的执行主编 Tracy Johnson，过去几年提供了友好且有益的指导。感谢 Jouve 公司的 Pavithra Jayapaul 以及培生出版社的产品编辑 Greg Dulles 为本书出版所做的出色工作。

早期版本

特别感谢以下诸位，他们在本书早期版本中提供了极大的帮助：

- William Barrett，圣何塞州立大学
- Scott Blackledge
- James Brink，太平洋路德大学
- Gerald Cahill，羚羊谷学院
- John Taylor

① 关于本书教辅资源，用书教师可向培生教育出版集团北京代表处申请，电话：010-57355169/ 57355171，电子邮件：service.cn@pearson.com。——编辑注

ASCII 控制字符

下表给出了按下控制键组合后生成的 ASCII 码。助记符和描述是指用于屏幕和打印机格式以及数据通信的 ASCII 函数。

ASCII 码 ^①	Ctrl-	助记符	描述	ASCII 码 ^①	Ctrl-	助记符	描述
00		NUL	空字符	10	Ctrl-P	DLE	换码
01	Ctrl-A	SOH	标题开始	11	Ctrl-Q	DC1	设备控制 1
02	Ctrl-B	STX	正文开始	12	Ctrl-R	DC2	设备控制 2
03	Ctrl-C	ETX	正文结束	13	Ctrl-S	DC3	设备控制 3
04	Ctrl-D	EOT	传输结束	14	Ctrl-T	DC4	设备控制 4
05	Ctrl-E	ENQ	查询	15	Ctrl-U	NAK	拒绝接收
06	Ctrl-F	ACK	确认	16	Ctrl-V	SYN	同步空闲
07	Ctrl-G	BEL	响铃	17	Ctrl-W	ETB	传输块结束
08	Ctrl-H	BS	退格	18	Ctrl-W	CAN	取消
09	Ctrl-I	HT	水平制表符	19	Ctrl-Y	EM	媒体结束
0A	Ctrl-J	LF	换行	1A	Ctrl-Z	SUB	替换
0B	Ctrl-K	VT	垂直制表符	1B	Ctrl-[ESC	退出
0C	Ctrl-L	FF	换页	1C	Ctrl-\	FS	文件分隔符
0D	Ctrl-M	CR	回车	1D	Ctrl-]	GS	分组符
0E	Ctrl-N	SO	不用切换	1E	Ctrl-^	RS	记录分隔符
0F	Ctrl-O	SI	启用切换	1F	Ctrl- ^②	UX	单元分隔符

① ASCII 码为十六进制。

② ASCII 码 1Fh 为 Ctrl- 连字符 (-)。

Alt 组合键

按住 Alt 键的同时按下其他键将会产生的十六进制扫描码：

键	扫描码	键	扫描码	键	扫描码
1	78	A	1E	N	31
2	79	B	30	O	18
3	7A	C	2E	P	19
4	7B	D	20	Q	10
5	7C	E	12	R	13
6	7D	F	21	S	1F
7	7E	G	22	T	14
8	7F	H	23	U	16
9	80	I	17	V	2F
0	81	J	24	W	11
-	82	K	25	X	2D
=	83	L	26	Y	15
		M	32	Z	2C

键盘扫描码

通过对键盘输入二次（第一次读键盘返回 0）调用 INT 16h 或 INT 21h 可获得键盘扫描码。所有扫描码均为十六进制：

功能键

键	正常	与 Shift 组合	与 Ctrl 组合	与 Alt 组合
F1	3B	54	5E	68
F2	3C	55	5F	69
F3	3D	56	60	6A
F4	3E	57	61	6B
F5	3F	58	62	6C
F6	40	59	63	6D
F7	41	5A	64	6E
F8	42	5B	65	6F
F9	43	5C	66	70
F10	44	5D	67	74
F11	85	87	89	8B
F12	86	88	8A	8C

键	单独使用	与 Ctrl 键组合
Home	47	77
End	4F	75
PgUp	49	84
PgDn	51	76
PrtSc	37	72
Left arrow	4B	73
Rt arrow	4D	74
Up arrow	48	8D
Dn arrow	50	91
Ins	52	92
Del	53	93
Back tab	0F	94
Gray +	4E	90
Gray -	4A	8E

十进制	⇒	1	16	32	48	64	80	96	112
↓	十六进制	0	1	2	3	4	5	6	7
0	0	null	▶	space	0	@	P	`	p
1	1	☺	◀	!	1	A	Q	a	q
2	2	☹	↕	"	2	B	R	b	r
3	3	♥	!!	#	3	C	S	c	s
4	4	♦	Π	\$	4	D	T	d	t
5	5	♣	§	%	5	E	U	e	u
6	6	♠	■	&	6	F	V	f	v
7	7	•	↕	'	7	G	W	g	w
8	8	■	^	(8	H	X	h	x
9	9	○	↓)	9	I	Y	i	y
10	A	◼	→	*	:	J	Z	j	z
11	B	♂	←	+	;	K	[k	{
12	C	♀	└	,	<	L	\	l	
13	D	♪	↔	-	=	M]	m	}
14	E	♫	▲	.	>	N	^	n	~
15	F	⚙	▼	/	?	O	_	o	Δ

十进制	⇒	128	144	160	176	192	208	224	240
	十六进制	8	9	A	B	C	D	E	F
0	0	Ç	É	á	⋮	L	⊥	α	≡
1	1	ü	æ	í	⊗	⊥	⊥	β	±
2	2	é	Æ	ó	⋮	⊥	π	Γ	≥
3	3	â	ô	ú		⊥	⊥	π	≤
4	4	ä	ö	ñ	⊥	-	⊥	Σ	∫
5	5	à	ò	Ñ	⊥	⊥	⊥	σ	∫
6	6	å	û	<u>a</u>	⊥	⊥	π	μ	÷
7	7	ç	ù	o	π	⊥	⊥	τ	≈
8	8	ê	ÿ	ı	π	⊥	⊥	Φ	o
9	9	ë	Ö	г	⊥	⊥	⊥	θ	•
10	A	è	Û	г		⊥	г	Ω	•
11	B	ï	ç	½	π	π	■	δ	√
12	C	î	£	¼	⊥	⊥	■	∞	n
13	D	ì	¥	i	⊥	=	⊥	φ	²
14	E	Ä	Pt	«	⊥	⊥	⊥	ε	▪
15	F	Å	f	»	г	⊥	■	∩	blank

目 录

Assembly Language for x86 Processors, Seventh Edition

出版者的话	
译者序	
前言	
第 1 章 基本概念	1
1.1 欢迎来到汇编语言的世界	1
1.1.1 读者可能会问的问题	2
1.1.2 汇编语言的应用	4
1.1.3 本节回顾	5
1.2 虚拟机概念	5
1.3 数据表示	7
1.3.1 二进制整数	7
1.3.2 二进制加法	8
1.3.3 整数存储大小	9
1.3.4 十六进制整数	10
1.3.5 十六进制加法	11
1.3.6 有符号二进制整数	12
1.3.7 二进制减法	13
1.3.8 字符存储	14
1.3.9 本节回顾	15
1.4 布尔表达式	16
1.4.1 布尔函数真值表	18
1.4.2 本节回顾	18
1.5 本章小结	19
1.6 关键术语	19
1.7 复习题和练习	20
1.7.1 简答题	20
1.7.2 算法基础	21
第 2 章 x86 处理器架构	23
2.1 一般概念	23
2.1.1 基本微机设计	23
2.1.2 指令执行周期	24
2.1.3 读取内存	25
2.1.4 加载并执行程序	26
2.1.5 本节回顾	26
2.2 32 位 x86 处理器	27
2.2.1 操作模式	27
2.2.2 基本执行环境	27
2.2.3 x86 内存管理	30
2.2.4 本节回顾	30
2.3 64 位 x86-64 处理器	30
2.3.1 64 位操作模式	31
2.3.2 基本 64 位执行环境	31
2.4 典型 x86 计算机组件	32
2.4.1 主板	32
2.4.2 内存	34
2.4.3 本节回顾	34
2.5 输入输出系统	34
2.5.1 I/O 访问层次	34
2.5.2 本节回顾	36
2.6 本章小结	36
2.7 关键术语	37
2.8 复习题	38
第 3 章 汇编语言基础	39
3.1 基本语言元素	39
3.1.1 第一个汇编语言程序	39
3.1.2 整数常量	40
3.1.3 整型常量表达式	41
3.1.4 实数常量	41
3.1.5 字符常量	42
3.1.6 字符串常量	42
3.1.7 保留字	42
3.1.8 标识符	43
3.1.9 伪指令	43
3.1.10 指令	44
3.1.11 本节回顾	46
3.2 示例：整数加减法	46
3.2.1 AddTwo 程序	46
3.2.2 运行和调试 AddTwo 程序	48
3.2.3 程序模板	52

3.2.4 本节回顾	52	第 4 章 数据传送、寻址和算术运算	73
3.3 汇编、链接和运行程序	53	4.1 数据传送指令	73
3.3.1 汇编-链接-执行周期	53	4.1.1 引言	73
3.3.2 列表文件	53	4.1.2 操作数类型	73
3.3.3 本节回顾	55	4.1.3 直接内存操作数	74
3.4 定义数据	55	4.1.4 MOV 指令	75
3.4.1 内部数据类型	55	4.1.5 整数的全零/符号扩展	76
3.4.2 数据定义语句	55	4.1.6 LAHF 和 SAHF 指令	77
3.4.3 向 AddTwo 程序添加一个 变量	56	4.1.7 XCHG 指令	78
3.4.4 定义 BYTE 和 SBYTE 数据	57	4.1.8 直接-偏移量操作数	78
3.4.5 定义 WORD 和 SWORD 数据	59	4.1.9 示例程序 (Moves)	79
3.4.6 定义 DWORD 和 SDWORD 数据	59	4.1.10 本节回顾	80
3.4.7 定义 QWORD 数据	60	4.2 加法和减法	81
3.4.8 定义压缩 BCD (TBYTE) 数据	60	4.2.1 INC 和 DEC 指令	81
3.4.9 定义浮点类型	61	4.2.2 ADD 指令	81
3.4.10 变量加法程序	61	4.2.3 SUB 指令	81
3.4.11 小端顺序	62	4.2.4 NEG 指令	82
3.4.12 声明未初始化数据	62	4.2.5 执行算术表达式	82
3.4.13 本节回顾	63	4.2.6 加减法影响的标志位	82
3.5 符号常量	63	4.2.7 示例程序 (AddSubTest)	85
3.5.1 等号伪指令	63	4.2.8 本节回顾	86
3.5.2 计算数组和字符串的大小	64	4.3 与数据相关的运算符和伪指令	87
3.5.3 EQU 伪指令	65	4.3.1 OFFSET 运算符	87
3.5.4 TEXTEQU 伪指令	66	4.3.2 ALIGN 伪指令	88
3.5.5 本节回顾	66	4.3.3 PTR 运算符	88
3.6 64 位编程	67	4.3.4 TYPE 运算符	89
3.7 本章小结	68	4.3.5 LENGTHOF 运算符	89
3.8 关键术语	69	4.3.6 SIZEOF 运算符	90
3.8.1 术语	69	4.3.7 LABEL 伪指令	90
3.8.2 指令、运算符和伪指令	70	4.3.8 本节回顾	90
3.9 复习题和练习	70	4.4 间接寻址	91
3.9.1 简答题	70	4.4.1 间接操作数	91
3.9.2 算法基础	71	4.4.2 数组	91
3.10 编程练习	71	4.4.3 变址操作数	92
		4.4.4 指针	93
		4.4.5 本节回顾	95
		4.5 JMP 和 LOOP 指令	95
		4.5.1 JMP 指令	96
		4.5.2 LOOP 指令	96
		4.5.3 在 Visual Studio 调试器中显示	

数组	97	5.5.1 Irvine64 链接库	139
4.5.4 整数数组求和	98	5.5.2 调用 64 位子程序	140
4.5.5 复制字符串	98	5.5.3 x64 调用规范	140
4.5.6 本节回顾	99	5.5.4 调用过程示例	141
4.6 64 位编程	99	5.6 本章小结	142
4.6.1 MOV 指令	99	5.7 关键术语	143
4.6.2 64 位的 SumArray 程序	100	5.7.1 术语	143
4.6.3 加法和减法	101	5.7.2 指令、运算符和伪指令	143
4.6.4 本节回顾	102	5.8 复习题和练习	143
4.7 本章小结	102	5.8.1 简答题	143
4.8 关键术语	104	5.8.2 算法基础	146
4.8.1 术语	104	5.9 编程练习	146
4.8.2 指令、运算符和伪指令	104	第 6 章 条件处理	148
4.9 复习题和练习	104	6.1 条件分支	148
4.9.1 简答题	104	6.2 布尔和比较指令	148
4.9.2 算法基础	106	6.2.1 CPU 状态标志	149
4.10 编程练习	107	6.2.2 AND 指令	149
第 5 章 过程	108	6.2.3 OR 指令	150
5.1 堆栈操作	108	6.2.4 位映射集	151
5.1.1 运行时堆栈 (32 位模式)	108	6.2.5 XOR 指令	152
5.1.2 PUSH 和 POP 指令	110	6.2.6 NOT 指令	153
5.1.3 本节回顾	112	6.2.7 TEST 指令	153
5.2 定义并使用过程	112	6.2.8 CMP 指令	154
5.2.1 PROC 伪指令	112	6.2.9 置位和清除单个 CPU 标志位	155
5.2.2 CALL 和 RET 指令	114	6.2.10 64 位模式下的布尔指令	155
5.2.3 过程调用嵌套	115	6.2.11 本节回顾	156
5.2.4 向过程传递寄存器参数	116	6.3 条件跳转	156
5.2.5 示例: 整数数组求和	116	6.3.1 条件结构	156
5.2.6 保存和恢复寄存器	118	6.3.2 Jcond 指令	156
5.2.7 本节回顾	119	6.3.3 条件跳转指令类型	157
5.3 链接到外部库	119	6.3.4 条件跳转应用	159
5.3.1 背景知识	119	6.3.5 本节回顾	163
5.3.2 本节回顾	120	6.4 条件循环指令	163
5.4 Irvine32 链接库	120	6.4.1 LOOPZ 和 LOOPE 指令	163
5.4.1 创建库的动机	120	6.4.2 LOOPNZ 和 LOOPNE 指令	164
5.4.2 概述	122	6.4.3 本节回顾	164
5.4.3 过程详细说明	123	6.5 条件结构	164
5.4.4 库测试程序	133	6.5.1 块结构的 IF 语句	165
5.4.5 本节回顾	139	6.5.2 复合表达式	167
5.5 64 位汇编编程	139		