

21世纪高等学校规划教材 | 计算机科学与技术



计算机组成与设计

王换招 陈妍 赵青苹 编著

清华大学出版社

013071505

21世纪高等学校规划教材 | 计算机科学与技术

TP303-43
129

计算机组成与设计



王换招 陈妍 赵青苹 编著



北航

C1680392

清华大学出版社
北京

TP303-43
129

202170810

内 容 简 介

本书系统地介绍了计算机的基本组成原理、内部工作机制以及相关设计方法。全书共分8章：计算机系统概论、指令系统、主存储器、存储系统、总线与输入输出系统、数据的表示与运算、中央处理器、控制器。

本书在讲述计算机一般原理的基础上，通过软硬件结合的方法力求达到原理与应用的结合。全书内容由浅入深，理论结合实际。每章之后均附有思考题和习题，便于读者通过思考和练习深入掌握所学知识。

本书可以作为高等院校计算机及相关专业“计算机组成原理”课程的教材，也可供从事计算机工作的工程技术人员参考。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目(CIP)数据

计算机组成与设计/王换招等编著. —北京：清华大学出版社，2013

21世纪高等学校规划教材·计算机科学与技术

ISBN 978-7-302-32884-1

I. ①计… II. ①王… III. ①计算机体系结构—高等学校—教材 IV. ①TP303

中国版本图书馆CIP数据核字(2013)第136389号

责任编辑：郑寅堃 薛 阳

封面设计：傅瑞学

责任校对：梁 毅

责任印制：刘海龙

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>, <http://www.wqbook.com>

地 址：北京清华大学学研大厦 A 座 邮 编：100084

社 总 机：010-62770175 邮 购：010-62786544

投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈：010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载：<http://www.tup.com.cn>, 010-62795954

印 装 者：清华大学印刷厂

经 销：全国新华书店

开 本：185mm×260mm

印 张：31

字 数：771千字

版 次：2013年8月第1版

印 次：2013年8月第1次印刷

印 数：1~2000

定 价：49.80元

产品编号：041861-01

清华大学出版社

北京

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和教学方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程”(简称“质量工程”),通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制定精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上。精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展,顺应并符合21世纪教学发展的规律,代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版

社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色精品教材包括:

- (1) 21世纪高等学校规划教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。
- (2) 21世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。
- (3) 21世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。
- (4) 21世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。
- (5) 21世纪高等学校规划教材·信息管理与信息系统。
- (6) 21世纪高等学校规划教材·财经管理与应用。
- (7) 21世纪高等学校规划教材·电子商务。
- (8) 21世纪高等学校规划教材·物联网。

清华大学出版社经过三十多年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

联系人:魏江江

E-mail: weijj@tup.tsinghua.edu.cn

前言

国内外关于“计算机组成原理”课程的教材有很多。但是,作为本课程的主讲教师,在选择和使用教材的过程中,时常会觉得很多教材都有些不尽如人意的地方。一般来说,国外教材内容更新紧跟时代技术潮流,细节展开翔实丰富。但是,原理性的总结归纳不足,并且由于课程体系的不同,往往在内容涵盖范围上与国内需求有较大的偏差,不太适合直接作为国内教学使用。国内教材的普遍特点是善于归纳总结,重点突出,语言规矩,结构规范。但是,内容更新较慢,与新技术、实际应用的结合不够紧密。

基于上述原因,编写本教材的基本出发点就设法在国内外教材间找到一个好的契合点,取长补短,力求达到原理与应用、硬件与软件、经典与流行等方面的有机结合,使本教材在具有易读、好理解等优势的前提下,更具启发性,能开拓读者的思路并引导读者深入学习更加深层的原理和方法。

“计算机组成原理”作为计算机核心专业课程,为学生建立完整的计算机系统概念,特别是硬件系统的工作原理、设计方法,以及软件和硬件之间的关系和接口等方面,将会起到非常重要的作用。为后续课程学习奠定良好且坚实的专业基础。

本教材主要讲授单台计算机的完整硬件系统的基本组成原理与内部运行机制。主要包括计算机硬件的基本组织方法,以及各部件的基本结构、工作原理和设计方法。在基本原理和方法讲解的基础上,从计算机程序设计者的视角出发,通过高级语言、汇编语言、机器语言程序等层次,力求使读者深入理解计算机最底层硬件的功能、结构和运行机制。


本教材共分8章,其中第1章、第2章、第5章、第7章和第8章主要由王换招编写,第3章和第4章主要由陈妍编写,第6章主要由赵青苹编写。本教材中的所有程序全部由唐亚哲编写和验证。张克旺在教材编写过程中提出了宝贵的修改建议。

本教材在编写过程中参考了国内外大量相关教材,从中吸取了大量宝贵经验。在此,向所有相关专家致以衷心的感谢。

由于作者水平有限,教材中难免有不妥之处,谨请专家和读者批评指正。

编者

2013年8月



目 录

第 1 章 计算机系统概论	1
1.1 计算机系统简介	1
1.1.1 计算机系统的组成	1
1.1.2 从应用程序透视计算机系统	2
1.1.3 计算机系统的层次结构	6
1.1.4 计算机组成和系统结构	9
1.2 计算机系统的发展与应用	10
1.2.1 计算机系统发展概况	10
1.2.2 计算机应用分类	15
1.3 计算机硬件的组成	16
1.3.1 冯·诺依曼计算机的特点	16
1.3.2 计算机硬件的基本组成	17
1.3.3 从程序运行透视计算机组成	18
1.4 计算机系统的性能	26
1.4.1 性能的定义	26
1.4.2 影响性能的因素	26
1.5 本书主要内容及组织结构	29
思考题与习题	30
第 2 章 指令系统	32
2.1 概述	32
2.2 指令系统的发展	33
2.2.1 指令系统的演变过程	33
2.2.2 CISC 与 RISC 指令系统的特点	34
2.3 指令系统的功能	36
2.3.1 指令系统的设计原则	36
2.3.2 数据类型	37
2.3.3 操作类型	44
*2.3.4 指令系统实例	46
*2.3.5 通过程序透视 CISC 和 RISC	49
2.4 指令格式	50

* 表示可选择讲述的内容。

2.4.1	指令的组成	50
2.4.2	指令字长	55
2.4.3	操作码扩展技术	56
*2.4.4	指令格式举例	58
2.5	寻址方式	60
2.5.1	寻址方式	60
*2.5.2	程序定位方式	69
附录 2A	IA-32 指令格式	72
附录 2B	ASCII	76
	思考题与习题	78
第 3 章	主存储器	83
3.1	主存储器概述	83
3.1.1	存储器的分类	83
3.1.2	主存储器与 CPU 的连接	86
3.1.3	主存储器的性能指标	86
3.2	半导体主存储器	87
3.2.1	主存储器的基本组成	88
3.2.2	SRAM	90
3.2.3	DRAM	94
*3.2.4	新型 DRAM	100
3.2.5	只读存储器和闪速存储器	102
3.2.6	存储器容量扩展及其与 CPU 的连接	107
*3.2.7	微处理器与存储器的连接举例	113
*3.3	高速主存储器	117
3.3.1	双端口存储器	118
3.3.2	多体交叉存储器	120
	思考题与习题	124
第 4 章	存储系统	127
4.1	存储系统的层次结构	127
*4.2	相联存储器	129
4.3	高速缓冲存储器	132
4.3.1	cache 的工作原理	132
4.3.2	cache 的设计要素	135
*4.3.3	cache 与主存储器间的调度举例	142
4.4	辅助存储器	146
4.4.1	概述	146
4.4.2	硬磁盘存储器	148

* 4.4.3 磁盘阵列	154
* 4.4.4 其他辅助存储器	158
* 4.5 虚拟存储器	162
4.5.1 虚拟存储器的基本原理	162
4.5.2 虚存存储器的地址映射及变换	165
4.5.3 存储保护	171
* 4.6 Pentium 4 系列机的存储器	172
4.6.1 Pentium 4 的 cache 结构	173
4.6.2 Pentium 4 的虚拟存储管理	174
思考题与习题	177
第 5 章 总线与输入输出系统	180
5.1 总线	180
5.1.1 总线的概念	180
5.1.2 总线的分类	181
5.2 总线的管理和控制	182
5.2.1 总线仲裁机制	183
5.2.2 总线通信方式	185
5.3 总线结构和标准	189
5.3.1 总线结构	189
5.3.2 总线标准及特性	190
* 5.3.3 总线举例	192
5.4 输入输出系统	195
5.5 I/O 设备	196
5.5.1 外设的基本组成和分类	196
* 5.5.2 输入设备	197
* 5.5.3 输出设备	202
5.6 I/O 接口	213
5.6.1 I/O 接口的作用和组成	213
5.6.2 I/O 接口的通信方式	214
5.7 I/O 数据传送控制方式	217
5.7.1 程序查询方式	217
5.7.2 程序中断方式	219
5.7.3 DMA 方式	227
思考题与习题	232
第 6 章 数据的表示与运算	236
6.1 计算机中表示信息的基本方法	236
6.1.1 计算机中常用的信息类型	236

151	* 6.1.2 非数值数据的表示	237
151	6.1.3 十进制数据的编码表示	243
151	* 6.1.4 数据校验码	246
151	6.2 定点数的表示	259
151	* 6.2.1 计算机中常用进位计数制及其相互转换	259
171	6.2.2 真值与机器数	263
171	6.2.3 常用机器码的表示	265
171	6.3 定点运算	272
171	6.3.1 运算部件的基本结构	272
171	6.3.2 定点加减运算	274
181	6.3.3 移位运算	281
181	6.3.4 定点乘法运算	288
181	6.3.5 定点除法运算	298
181	6.3.6 阵列乘除法器	307
181	6.3.7 十进制运算	311
181	6.3.8 基本的逻辑运算	315
181	6.4 定点运算器的实现	317
181	6.4.1 加法器的进位技术	317
181	6.4.2 算术逻辑单元	323
181	6.4.3 定点运算器的基本结构	328
191	6.5 浮点数的表示与运算	336
191	6.5.1 浮点数的基本格式	336
191	6.5.2 浮点阶的移码表示	337
191	6.5.3 浮点表示法	339
191	* 6.5.4 IEEE754 浮点标准	347
191	6.5.5 规格化浮点加减运算	349
191	6.5.6 规格化浮点乘除运算	353
191	6.5.7 浮点运算的实现	358
191	* 6.5.8 C 语言数据操作的机内透视	360
191	* 6.5.9 提高运算器性能的措施	363
191	思考题与习题	365
191	第 7 章 中央处理器	370
191	7.1 CPU 的功能和组成	370
191	7.1.1 CPU 的功能	371
191	7.1.2 CPU 的基本结构	372
191	7.2 CPU 的设计方法	373
191	7.2.1 CPU 的设计过程	373
191	7.2.2 寄存器传输语言	374

7.3	CPU 数据通路的结构和组成	376
7.3.1	数据通路的操作分析	376
7.3.2	数据通路的基本部件	378
7.3.3	CPU 结构分类	381
7.3.4	目标指令集假设	383
7.4	中断系统	384
7.4.1	中断源分类	384
7.4.2	中断响应	385
7.5	单周期 CPU 数据通路	387
7.5.1	数据通路	388
7.5.2	指令周期流程	392
7.5.3	指令周期与 CPU 性能	394
*7.5.4	支持内部中断的数据通路	397
7.6	多周期数据通路	398
7.6.1	数据通路的基本设计方法	399
7.6.2	分散互连结构	399
7.6.3	单总线结构	406
7.6.4	双总线和三总线结构	409
7.6.5	指令周期及机器性能	413
7.7	指令流水处理器	414
7.7.1	指令流水原理	414
7.7.2	指令流水线性能	421
7.7.3	流水 CPU 数据通路	422
	思考题与习题	429
第 8 章	控制器	435
8.1	控制器的基本结构和设计方法	435
8.1.1	控制器的功能	435
8.1.2	控制器的组成和设计方法	437
8.2	计算机的控制方式	438
8.2.1	计算机中的时序系统	438
8.2.2	控制方式	439
8.2.3	时序信号产生器	442
8.3	组合逻辑控制器	443
8.3.1	硬布线控制器	444
8.3.2	门阵列控制器	446
8.4	微程序控制器	453
8.4.1	微程序控制器的设计思想	453
8.4.2	微程序控制单元的基本结构	453

8.4.3	微指令格式设计	455
8.4.4	微指令格式和微程序设计实例	461
8.4.5	微程序控制单元运行实例	468
*8.4.6	微程序控制单元的操作定时	469
*8.4.7	动态微程序设计	470
*8.5	混合式控制器	471
8.6	流水线控制器	472
8.6.1	流水线控制器的基本结构	472
*8.6.2	避免冒险的流水线控制器	474
	思考题与习题	477
	参考文献	481

第1章

计算机系统概论

本章从计算机系统的概念入手,主要介绍计算机系统中硬件部分的组成及其基本工作原理,并明确本教材讨论的主要内容。简单介绍计算机系统的性能,为后续各章的深入讨论奠定基础。

1.1 计算机系统简介

1.1.1 计算机系统的组成

现代计算机的用途非常广泛,但就其本质来说,计算机就是按照人们给出的指令执行具体操作来解决各种问题的机器。计算机中的器件、线路和设备是指令的执行实体。描述如何完成一个确定任务的指令序列称为程序,那么计算机就是能够执行各种各样程序的电子设备。

所谓计算机硬件,即组成计算机各种实际装置的总称,由各类光、电、机等器件或设备的实物组成,例如处理器芯片、内存条以及各类外部设备等。硬件是指令的接受者和执行者,它是人们可以看得见摸得着的有形实体。虽然,计算机硬件只能识别和执行极其简单的低级指令(机器指令),但是,它是软件运行的载体。

所谓计算机软件,即由人们事先编制的具有各类特殊功能的程序和相关数据组成。通常以某种特殊形式存放在存储介质中,运行时以电信号形式在硬件实体上流动。软件是指令的产生者和发布者。人们只能看到存放软件的载体,例如磁盘、光盘等,软件本体是看不见摸不着的。只有使软件在计算机中运行,计算机才能真正发挥其作用。

计算机完成任务要由硬件和软件两部分共同实现。所以,硬件和软件构成了计算机系统的两个基本要素。硬件是计算机系统的物质基础,软件就好像是计算机系统的灵魂。硬件和软件相辅相成,相互依存,缺一不可。硬件和软件的性能共同决定计算机系统的性能。

计算机的软件通常又可以分为两大类:系统软件和应用软件。

系统软件又称为系统程序,主要用来管理整个计算机系统,监视服务使系统资源得到合理调度,高效运行。它包括标准程序库、语言处理程序(如编辑器、编译器、汇编器)、操作系统、服务程序(如诊断程序、调试程序、连接程序等)、数据库管理系统、网络软件等。

应用软件又称为应用程序,它是用户根据具体任务需要所编制的各种程序,如科学计算程序、数据处理程序、过程控制程序、事务管理程序等。

图 1-1 是计算机系统的基本组成。其中,硬件是计算机系统的核心,其外围是系统软件,计算机系统的最外层是应用软件。

1.1.2 从应用程序透视计算机系统

本小节从用户使用计算机的角度,通过一个简单高级语言(C语言)程序从编制到转换成机器代码的过程,简单介绍计算机系统中各类软件的作用。1.3节将对计算机硬件按照功能模块进行划分,然后通过机器码程序在硬件上的运行过程,初步阐述各部分硬件的功能和它们之间的关系。通过这个实例希望读者能对计算机系统有一个整体的认识。

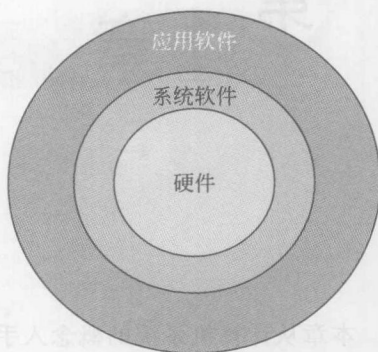


图 1-1 计算机系统的基本构成

用计算机解决一个实际问题通常包含四个步骤:编程前准备、编制程序、转换程序和运行程序。

1. 编程前准备

在应用计算机解决某种具体科学问题的实际应用中,往往会遇到许多复杂的数学问题,这就要求在编制计算机程序前,先由人工完成一些必要的准备工作。这些工作主要包括建立数学模型和确定计算方法。实际上就是把待解决的问题用数学语言描述出来。

2. 编制程序

程序是为实现特定目标或解决特定问题而用计算机语言编写的命令集合。采用当代计算机解题通常用高级语言(如 C/C++、Java 等)编写程序。

假设已经用某种编辑器(如 editplus、UNIX/Linux Vi 等)编写了一个 C 语言源程序 demo-1-1.c,该程序存储于硬盘文件中。源程序如图 1-2 所示。

```
//demo-1-1.c 输入变量初始值,求解某个表达式的值,将结果输出
int i,j,k,m,n;           //定义 5 个整型变量
main()
{
    scanf("%d%d%d%d", &i, &j, &m, &n); //从键盘输入 4 个整数
    k = i + j * m / n;           //计算表达式的值 k
    printf("%d", k);           //在屏幕上显示计算结果
}
```

图 1-2 demo-1-1.c

3. 转换程序

目前,通用计算机不能直接执行高级语言程序,所以,要把高级语言程序翻译成计算机可执行的机器语言程序。把高级语言程序翻译成机器语言程序有两种方法:解释和编译。解释是一边翻译一边执行,而编译是翻译完后再执行。由于采用解释方法执行程序效率低,所以,大多数高级语言采用编译方法。但是,编译方法并不是一步转换完成的,而是通过编

译、汇编和连接三个过程实现的。

编译是从源代码(高级语言)到能直接被计算机或虚拟机执行的目标代码(低级语言或机器语言)的翻译过程。通常,编译器将源程序(如 C 程序)转换成一种符号形式的汇编语言程序。汇编语言比高级语言更接近机器语言,但是高级语言程序比汇编语言程序使用更少的代码行,所以编程效率更高。

汇编是把汇编语言程序翻译成与之等价的机器语言程序的过程。汇编器输入汇编语言源程序,输出的目标文件包括机器语言指令、数据和将指令正确放入内存所需要的信息。

链接是对每一个独立汇编的机器语言程序地址进行绑定并分配相对地址,最后把这些机器语言程序与标准库程序“拼接”在一起,产生一个可以在计算机上运行的可执行文件。链接的方法有两种:静态链接和动态链接。静态链接产生的可执行程序可以多次加载执行,其特点是简单易用。但是,当库程序版本提高后必须重新链接,而且由于加载了大量可能不用的库程序代码,浪费了大量的内存空间。动态链接是每当程序运行时才进行链接和加载,并且不同程序可以共享同一个库程序副本。

图 1-3 描述了把 C 语言程序转换成可执行程序的过程。

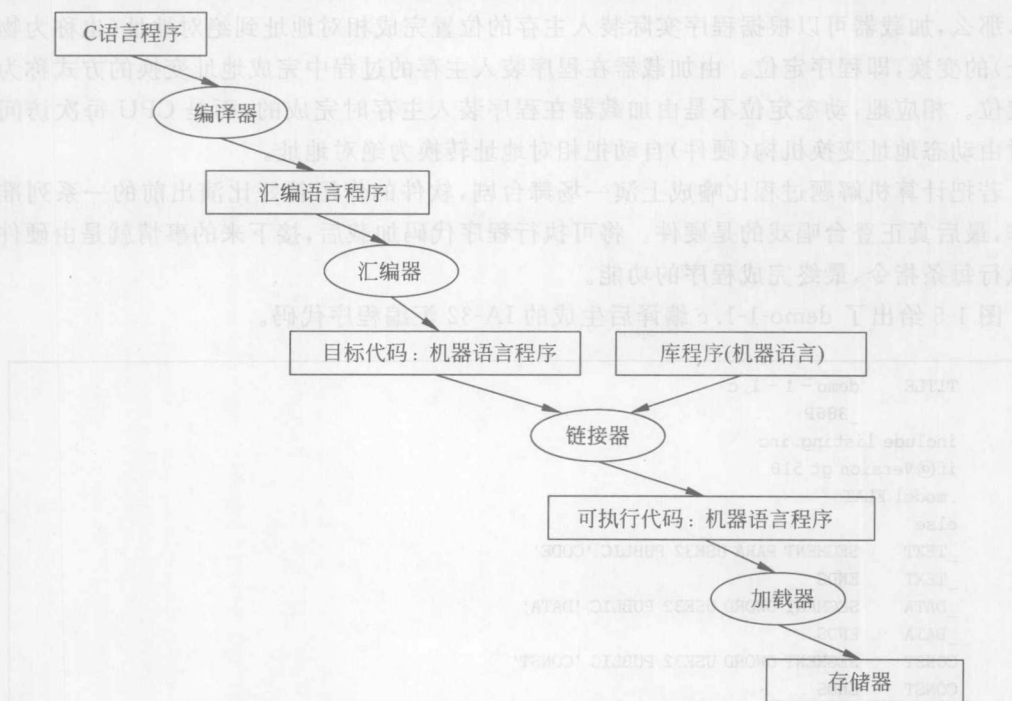


图 1-3 C 程序转换过程

图 1-4 给出 demo-1-1.c 的编译、汇编和链接过程。本书中使用 Visual C++ 6.0 来编译和链接基于 IA-32 指令系统的 C 程序。为清晰起见,使用命令行方式进行编译和链接。通过 cl 命令把 C 语言源程序转换成对应的 asm 汇编程序,然后通过 ml 命令把 asm 汇编程序汇编成 obj 目标文件,最后通过 link 程序把 obj 目标文件和相关的库文件一起链接成 exe 可执行程序。

```

D:\Program Files\Microsoft Visual Studio\VC98\Bin>cl /c /FA demo-1-1.c
Microsoft (R) 32-bit C/C++ Optimizing Compiler Version 12.00.8168 for 80x86
Copyright (C) Microsoft Corp 1984-1998. All rights reserved.

demo-1-1.c

D:\Program Files\Microsoft Visual Studio\VC98\Bin>ml /c /Zi /coff demo-1-1.asm
Microsoft (R) Macro Assembler Version 6.14.8444
Copyright (C) Microsoft Corp 1981-1997. All rights reserved.

Assembling: demo-1-1.asm

D:\Program Files\Microsoft Visual Studio\VC98\Bin>link demo-1-1.obj msucrt.lib
Microsoft (R) Incremental Linker Version 6.00.8168
Copyright (C) Microsoft Corp 1992-1998. All rights reserved.

D:\Program Files\Microsoft Visual Studio\VC98\Bin>

```

图 1-4 一个 C 程序的编译、汇编和链接过程

4. 运行程序

通常,转换后的可执行程序以文件形式存储于磁盘中,用户运行该程序时,由操作系统中的加载器将可执行程序调入内存并启动运行。

汇编器生成的目标代码一般是从 0 地址开始编排的(采用所谓的相对地址或者逻辑地址),那么,加载器可以根据程序实际装入主存的位置完成相对地址到绝对地址(也称为物理地址)的变换,即程序定位。由加载器在程序装入主存的过程中完成地址变换的方式称为静态定位。相应地,动态定位不是由加载器在程序装入主存时完成的,而是 CPU 每次访问主存时由动态地址变换机构(硬件)自动把相对地址转换为绝对地址。

若把计算机解题过程比喻成上演一场舞台剧,软件的作用就好比演出前的一系列准备工作,最后真正登台唱戏的是硬件。将可执行程序代码加载后,接下来的事情就是由硬件依次执行每条指令,最终完成程序的功能。

图 1-5 给出了 demo-1-1.c 编译后生成的 IA-32 汇编程序代码。

```

TITLE    demo-1-1.c
        _386P
include listing.inc
if@Version gt 510
.model FLAT
else
_TEXT   SEGMENT PARA USE32 PUBLIC 'CODE'
_TEXT   ENDS
_DATA   SEGMENT DWORD USE32 PUBLIC 'DATA'
_DATA   ENDS
CONST   SEGMENT DWORD USE32 PUBLIC 'CONST'
CONST   ENDS
_BSS    SEGMENT DWORD USE32 PUBLIC 'BSS'
_BSS    ENDS
_TLS    SEGMENT DWORD USE32 PUBLIC 'TLS'
_TLS    ENDS
FLAT    GROUP _DATA, CONST, _BSS
        ASSUME CS:FLAT, DS:FLAT, SS:FLAT
endif
_DATA   SEGMENT
COMM    _i:DWORD

```

图 1-5 demo-1-1.asm

COMM	_j:DWORD				
COMM	_k:DWORD				
COMM	_m:DWORD				
COMM	_n:DWORD				
_DATA	ENDS				
PUBLIC	_main				
EXTRN	_scanf:NEAR				
EXTRN	_print:NEAR				
_DATA	SEGMENT				
\$ SG37	DB	' % 4 % 4 % 4 % 4 ',00H			
	ORG \$ + 3				
\$ SG39	DB	' % d ',00H			
_DATA	ENDS				
_TEXT	SEGMENT				
_main	PROC NEAR				
	;File 200.c				
	;Line 3				
	PUSH	ebp			
	MOV	ebp,esp			
	;Line 4				
	PUSH	OFFSET FLAT:_n	//压入参数到堆栈中		
	PUSH	OFFSET FLAT:_m	//压入参数到堆栈中		
	PUSH	OFFSET FLAT:_j	//压入参数到堆栈中		
	PUSH	OFFSET FLAT:_i	//压入参数到堆栈中		
	PUSH	OFFSET FLAT:\$ SG37	//压入参数到堆栈中(print的格式串)		
	CALL	_scanf	//调用 scanf 函数		
	ADD	ESP,20	//参数出栈,5个参数,每个4B,共20B		
	; Line 5				
	MOV	EAX,DWORD PTR_j	//把j的值取到 eax 寄存器中		
	IMUL	EAX,DWORD PTR_m	//j和m做乘法,结果存放到EAX中		
	CDQ		//符号扩展,把EAX中的数符号扩展到EDX:EAX		
	IDIV	DWORD PTR_n	//j和m的乘积除以n,结果到EAX中		
	MOV	ECX,DWORD PTR_i	//把i装入到ECX中		
	ADD	ECX,EAX	//EAX和ECX相加(i和j,m乘积除以n的结果相加)		
	MOV	DWORD PTR_k,ECX	//加法结果放到k中		
	;Line 6				
	MOV	EDX,DWORD PTR_k	//k值存放到EDX中		
	PUSH	EDX	//k值压栈		
	PUSH	OFFSET FLAT:\$ SG39	//格式串地址压栈		
	CALL	_printf	//调用 printf 函数		
	ADD	ESP,8	//k值和格式串地址出栈,两个参数,共8B		
	;Line 7				
	POP	EBP			
	RET	0			
_main	ENDP				
_TEXT	ENDS				
END					

图 1-5 (续)

图 1-6 给出了汇编、链接后产生的可执行程序(demo-1-1.exe)的机器指令代码。假定将该可执行程序的局部代码加载到首地址为 00401000h 的主存单元。图中第一列是各条机器指令所存放的主存单元地址,第二列是机器指令码,第三列是对应的汇编代码。