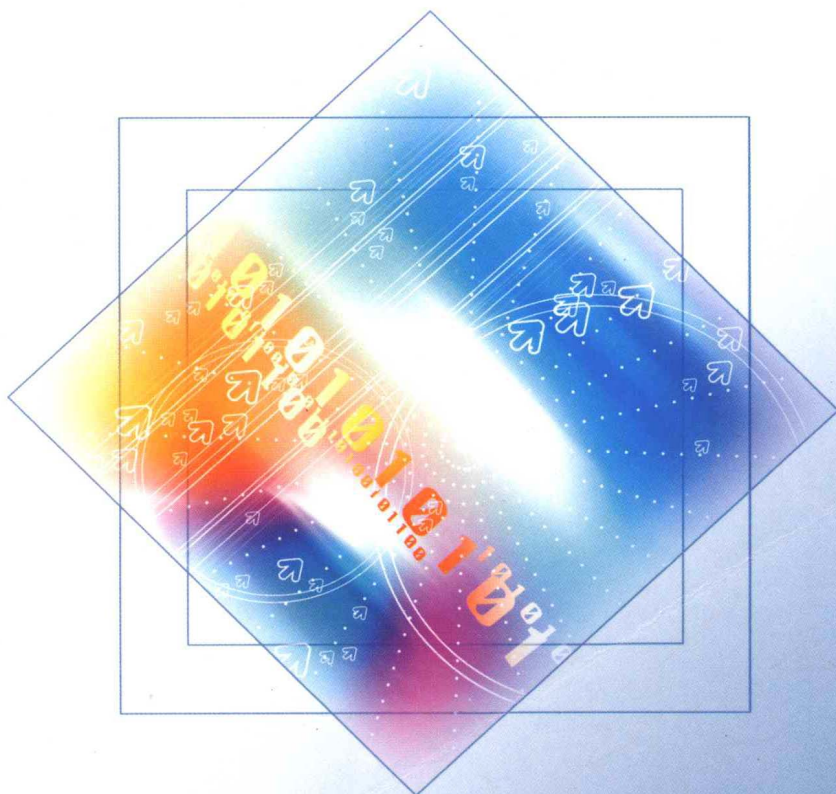


21 世纪高等院校计算机系列教材

Computer

□ 苏帆 唐永兴 吴保荣 胡杰 主编

汇编语言程序设计



华中科技大学出版社

21 世纪高等院校计算机系列教材

汇编语言程序设计

主 编 苏 帆 唐永兴 吴保荣 胡 杰

副主编 姚 娟 曾 武 温红艳

华中科技大学出版社

图书在版编目(CIP)数据

汇编语言程序设计/苏帆 等主编
武汉:华中科技大学出版社,2005年2月
ISBN 7-5609-3341-6

- I. 汇…
II. ①苏… ②唐… ③吴… ④胡…
III. 汇编语言-程序设计-教材
IV. TP312

汇编语言程序设计

苏帆 等主编

责任编辑:彭保林 曾 光
责任校对:吴 晗

封面设计:刘 卉
责任监印:熊庆玉

出版发行:华中科技大学出版社
武昌喻家山 邮编:430074 电话:(027)87557437

录 排:北京搜获科技有限公司
印 刷:武汉市新华印刷有限责任公司

开本:787×1092 1/16
版次:2005年2月第1版
ISBN 7-5609-3341-6/TP·556

印张:19.25
印次:2005年2月第1次印刷

字数:440 000
定价:28.00 元

(本书若有印装质量问题,请向出版社发行部调换)

内 容 简 介

本书主要阐述 80x86 汇编语言的程序设计方法和技术。全书共 10 章,第 1 章为基础知识部分;第 2 章和第 3 章为编程工具部分,主要内容为 80x86 的寻址方式与指令系统,以及包括伪指令在内的汇编语言程序格式;第 4 章至第 7 章介绍编程方法,内容包括顺序、分支、循环、子程序等基本程序结构,程序设计的基本方法和技术,多模块连接技术,宏汇编技术,以中断为主的输入输出程序设计方法,以及 BIOS 和 DOS 系统功能调用;第 8 章为实际应用部分,说明图形显示、发声和键盘录入技术,以及汇编语言与 C 语言接口的方法;第 9 章和第 10 章为知识拓展部分,主要介绍了分段、分页管理与多任务保护机制,实模式与保护模式切换实例,虚拟 86 方式的使用程序,多媒体汇编 MMX 和 SSE 指令及其编程设计与优化处理等。全书提供了大量程序实例,每章后均附有习题。

本书适用于高等院校以及大、中专院校作为“汇编语言程序设计”课程的教材,也可作为计算机工作者学习汇编语言的自学参考书,只要具有高级语言程序设计基础的读者,都可通过本书学习和掌握汇编语言程序设计技术。

“21 世纪高等院校计算机系列教材”丛书编委会

主任 何炎祥

委员（按姓氏拼音排序）：

戴光明	都志辉	桂 超	金先级
柯敏毅	李康顺	李克清	李禹生
刘腾红	卢强华	陆 迟	吕顺营
沈海波	石 清	王江晴	王伟军
王 忠	叶骏民	余敦辉	湛为芳

序 言

21世纪是信息时代，以计算机为核心的信息技术是21世纪科技发展的大趋势。作为计算机专业人才培养基地的大学计算机专业和相关专业，如何适应这种发展，培养出符合时代要求和社会欢迎的人才，是近年来计算机教育界讨论的热门话题，也是我们长期思考并努力探索的课题。

教材是人才培养的基础。在华中科技大学出版社的委托下，我们组织了有关高等院校的部分专家、教授共同编写了这套“面向21世纪计算机系列教材”，以期在适应21世纪的教材建设方面做出自己的努力。由于计算机行业发展日新月异，“21世纪计算机系列教材”编委会将负责系列教材的选题、每本教材大纲的编写和审定，以及教材、教学辅导书和课件的修订、更新等工作，以确保教材的正确性和先进性，使这套教材努力走在同类教材的前列。

这套系列教材包括计算机专业课和部分专业基础课教材，以及与之配套的实践课教材和教学辅导书等等。

我们希望这套教材具有以下特点：

1. 注重基础性和先进性的结合。计算机学科的一个显著特点就是知识和技术更新快，这对教学内容、课程知识结构的选取和组织提出了新的要求。我们把编写的重点放在基础知识、基本技能和基本方法上，希望在提高学生的理论素养和分析问题、解决问题的能力同时，注重介绍新的技术和方法，以拓展学生的知识面，激发他们学习的积极性和创新意识。

2. 注重理论性与应用性的结合。良好的理论素养是应用的前提，而掌握理论的目的就是为了更好的应用。在教材的编写过程中，我们注意理论的系统性，在讲深讲透主要知识的基础上，融理论性和应用性于一体，注意基本方法的讲授，以培养学生应用理论和技术的能力。

3. 注重时代性和实用性的结合。力求精简旧的知识点，增加新的知识点，体现教材的时代特征。而且充分考虑一般高校目前所拥有的师资条件和教学设备，注重教材的实用性。

4. 注重科学性与通俗性的结合。概念、原理、新技术的阐述力求准确、精练；写作上尽量通俗易懂、深入浅出、图文并茂，增强可读性，便于学生自学。

5. 网络技术辅助教学。针对本系列教材我们开发有专门的网站(<http://www.hzpress.org>)、课件发布演示系统和考试系统等，以便为任课老师的教学提供更便捷、更全面的服务，并将通过网站开展各种形式的教材网上专家答疑、内容修订发布、课件定期升级等活动，以与读者随时互动，为读者提供立体化的服务。

教学改革是需要不断探索的课题。要达到以上目标，还需要不断地努力实践和完善。欢迎使用这套教材的教师、学生和其他读者提出宝贵意见。

最后，对参加这套教材编写的所有作者，对为这套教材的编写提供支持的有关学校、院系的领导和老师表示诚挚的谢意！感谢华中科技大学出版社为本系列教材的出版所付出的辛勤劳动！

教材编委会主任 **何炎祥**

(教授、博导、武汉大学计算机学院院长)

前 言

汇编语言是一种能充分发挥和利用计算机硬件特性的语言，尤其适用于对时空效率要求较高、与机器硬件密切相关的高性能软件的开发。汇编语言面向机器，能够为程序员提供最直接操纵机器硬件系统的途径，利用它可以编写出在“时间”和“空间”两个方面最具效率的程序，例如，操作系统的核心代码，要求有快速响应的实时系统等。

随着计算机运行速度的日益提高和内存容量的不断扩大，汇编语言的许多优势似乎已不再突出。然而，汇编语言可以实现高级语言难以胜任甚至无法完成的工作，所以，至今仍是高级语言所无法替代的。掌握汇编语言知识，对于提高程序设计能力、加深对计算机系统的理解，无疑是非常重要的。很难想象一个不具备汇编语言知识的人能成为高水平的程序员，能设计出高质量的程序来。对于计算机专业人员来说，学习汇编语言是必要的。

《汇编语言程序设计》是高等院校计算机专业的一门重要技术基础课。是必修的核心课程之一，是《操作系统》、《编译原理》和《微机原理与接口技术》等其他核心课程的基础。该课程对于训练学生掌握程序设计技术，熟悉上机操作和程序调试技术都有重要作用。此外，《汇编语言程序设计》也是其他相关专业的必修或选修课。

目前，国内最广泛使用的 PC 系列机(包括兼容机)，都以 Intel 的 80x86 系列微处理器或者兼容的微处理器为 CPU。在 Intel 的 80x86 家族中，16 位的 8086 / 8088 是基础，实现了以分段方式管理存储器；32 位的 80386 是高档微处理器的里程碑，实现了支持多任务的保护工作方式；基于 MMX 技术的 Pentium 是新一代的微处理器，实现了对多媒体处理的支持。

本书以 8086 / 8088 为基础，面向 Pentium 等新一代微处理器来讲解汇编语言程序设计的一般概念、基本技术和常用技巧，介绍宏和模块化程序设计的技术方法，讲解保护模式编程的相关概念、编程技术及实现细节。

全书由“基础理论、编程工具、编程方法、实际应用和知识拓展”五部分，共 10 章组成。第 1 章为基础理论部分：包括基础知识，计算机组成及基本原理；第 2 章和第 3 章介绍编程工具：包括指令系统、寻址方式、伪操作和汇编语言格式；第 4 章至第 7 章说明编程方法：包括顺序、分支、循环、子程序等基本程序结构，宏和条件汇编等高级汇编技术，中断等输入输出程序设计方法，BIOS 和 DOS 系统功能调用方法，以及多个模块的连接技术；第 8 章为实际应用：包括图形显示、发声和键盘录入技术，以及汇编语言与 C 语言接口的方法；第 9 章和第 10 章为知识拓展部分：主要介绍了分段、分页管理与多任务保护机制，实模式与保护模式切换实例，虚拟 86 方式的使用程序，多媒体汇编 MMX 和 SSE 指令及其编程设计与优化处理等。

本书不求面面俱到，但求重点讲清。对于难以理解、容易混淆的内容力求给予明确说明。全书提供了大量程序实例，每章均附有习题，便于读者复习及检查学习效果。同时为了适应各种类型院校的不同要求，各章之间相互配合而又自成体系，易于为不同类型院校

按其要求适当选学，其中，带星号的章节可选修，教师可根据自己的教学计划做相应取舍，所以本教材的使用范围广。

全书共 10 章，由吴保荣编写第 1 章，苏帆编写第 2 章和第 3 章，胡杰编写第 4 章，曾武编写第 5 章和第 8 章，姚娟编写第 6 章，温红艳编写第 7 章，唐永兴编写第 9 章和第 10 章。在编写过程中，我们参考了有关的资料和教材，在此对他们表示感谢。

由于时间有限，书中难免有遗漏和不妥之处，恳请读者批评指正。

编 者

2004 年 12 月

目 录

第 1 章 汇编语言基础知识	(1)
1.1 计算机系统概述	(1)
1.1.1 微机系统的基本组成	(1)
1.1.2 计算机中数据信息的类型和表示	(4)
1.2 计算机语言	(11)
1.2.1 机器语言	(11)
1.2.2 汇编语言	(11)
1.2.3 高级语言	(12)
1.3 微处理器 8086/8088 简介	(13)
1.3.1 8088 CPU 的编程结构	(13)
1.3.2 8088 CPU 寄存器组	(15)
1.3.3 8088 CPU 存储器组织	(18)
1.4 汇编语言的上机操作	(22)
1.4.1 软件环境	(22)
1.4.2 汇编语言源程序上机操作步骤	(23)
1.4.3 MASM 回顾和发展	(26)
1.4.4 调试工具 DEBUG	(28)
习题 1	(33)
第 2 章 8086/8088 寻址方式和指令系统	(34)
2.1 8086/8088 的寻址方式	(34)
2.1.1 立即寻址方式	(34)
2.1.2 寄存器寻址方式	(35)
2.1.3 直接寻址方式	(35)
2.1.4 寄存器间接寻址方式	(36)
2.1.5 寄存器相对寻址方式	(37)
2.1.6 基址加变址寻址方式	(38)
2.1.7 相对基址加变址寻址方式	(39)
2.2 8086/8088 指令系统	(40)
2.2.1 8086/8088 指令概述	(40)
2.2.2 数据传送指令	(41)
2.2.3 算术运算指令	(47)
2.2.4 逻辑运算指令	(53)
2.2.5 移位指令	(55)
2.2.6 控制转移指令	(59)
2.2.7 标志位操作指令	(65)
2.2.8 处理机控制指令	(66)

习题 2	(67)
第 3 章 汇编语言程序格式	(73)
3.1 汇编语言语句	(73)
3.1.1 语句的种类和格式	(73)
3.1.2 表达式	(74)
3.1.3 特殊操作符	(77)
3.2 汇编语言伪指令	(80)
3.2.1 符号定义语句	(80)
3.2.2 数据定义语句	(83)
3.2.3 段定义语句	(85)
3.3 地址计数器	(88)
3.4 汇编语言源程序的结构	(89)
3.4.1 源程序的基本结构	(89)
3.4.2 基本的 DOS 功能调用	(91)
习题 3	(94)
第 4 章 汇编语言结构化程序设计	(96)
4.1 汇编语言程序设计的基本步骤	(96)
4.2 顺序程序设计	(96)
4.2.1 顺序程序的结构	(97)
4.2.2 顺序程序设计	(97)
4.3 分支程序设计	(98)
4.3.1 分支程序的结构	(98)
4.3.2 双分支程序设计	(99)
4.3.3 多分支程序设计	(100)
4.4 循环程序设计	(103)
4.4.1 循环程序基本结构	(103)
4.4.2 循环程序设计	(104)
4.4.3 循环程序的控制方法	(111)
4.5 串操作	(114)
4.5.1 字符串操作指令及重复前缀	(114)
4.5.2 字符串操作示例	(118)
习题 4	(125)
第 5 章 汇编语言模块化程序设计	(127)
5.1 子程序设计	(127)
5.1.1 子程序的设计方法	(127)
5.1.2 子程序与主程序的参数传递	(131)
5.1.3 嵌套子程序	(137)
5.2 模块化程序设计	(140)
5.2.1 段的完整定义	(140)
5.2.2 段的简化定义	(144)

5.2.3 模块间的通信	(145)
习题 5	(155)
第 6 章 高级汇编语言技术	(157)
6.1 宏汇编	(157)
6.1.1 宏的定义和调用	(157)
6.1.2 宏的参数	(159)
6.1.3 与宏有关的伪指令	(161)
6.1.4 宏与子程序	(163)
*6.2 重复汇编	(163)
6.2.1 重复伪指令	(164)
6.2.2 不定重复伪指令	(164)
*6.3 条件汇编	(165)
*6.4 数据结构	(169)
6.4.1 结构类型的说明	(169)
6.4.2 结构变量的定义	(169)
6.4.3 结构变量及其字段的引用	(170)
*6.5 数据记录	(171)
6.5.1 记录类型的说明	(171)
6.5.2 记录变量的定义	(171)
6.5.3 记录变量的引用和记录操作符	(171)
习题 6	(172)
第 7 章 I/O 和中断系统	(174)
7.1 I/O 的基本概念	(174)
7.1.1 I/O 指令和 I/O 端口编址	(174)
7.1.2 数据传送方式	(176)
7.2 中断系统	(183)
7.2.1 中断的基本概念	(183)
7.2.2 中断指令	(186)
7.2.3 中断向量及分类	(186)
7.3 DOS 中断与系统功能调用	(189)
7.3.1 DOS 功能调用的方法(包括文件)	(189)
7.3.2 DOS 基本 I/O 功能调用	(190)
7.4 BIOS 中断调用	(202)
7.4.1 BIOS 中断调用方法	(203)
7.4.2 常用 BIOS 功能调用	(203)
习题 7	(207)
第 8 章 应用程序设计	(210)
8.1 应用程序实例	(210)
8.1.1 发声程序设计	(210)
8.1.2 彩色图形程序设计	(213)

8.1.3 键盘录入数据的转换与显示	(215)
8.2 汇编语言与 C 语言的接口	(218)
8.2.1 嵌入汇编	(218)
8.2.2 汇编语言与 C 语言的混合编程	(220)
习题 8	(222)
第 9 章 保护虚拟机制及其程序设计	(224)
9.1 保护程序设计基础	(224)
9.1.1 IA-32 处理器寄存器组	(224)
9.1.2 保护模式下存储器管理机制	(229)
9.1.3 保护机制	(242)
9.2 特殊指令集	(245)
9.2.1 实模式下可执行的 0 特权级指令	(245)
9.2.2 实模式下可执行的任意特权级指令	(247)
9.2.3 只能在保护模式下执行的指令	(249)
9.2.4 特权指令	(251)
9.3 实模式与保护模式切换	(252)
9.3.1 模式切换方法	(252)
9.3.2 模式切换程序设计示例	(254)
9.4 虚拟 86 方式	(258)
9.4.1 V86 任务	(259)
9.4.2 V86 任务中的保护	(259)
9.4.3 V86 方式的进入与退出	(260)
习题 9	(263)
第 10 章 多媒体汇编指令及其编程	(264)
10.1 MMX 技术编程环境	(264)
10.1.1 MMX 寄存器	(265)
10.1.2 MMX 数据类型	(266)
10.1.3 MMX 指令集	(266)
10.2 MMX 程序设计	(275)
10.2.1 确认微处理器支持 MMX 指令	(275)
10.2.2 MMX 指令与浮点指令的混合使用	(275)
10.2.3 MMX 程序的优化	(276)
10.2.4 MMX 程序设计示例	(276)
10.3 SSE 编程环境	(278)
10.3.1 SSE 寄存器	(278)
10.3.2 SSE 数据类型	(280)
10.3.3 SSE 指令集	(280)
习题 10	(294)
参考文献	(295)

第 1 章 汇编语言基础知识

程序设计语言是开发软件的工具，它的发展经历了由低级语言到高级语言的过程，汇编语言是一种面向机器的低级程序设计语言。

汇编语言以助记符的形式表示每一条计算机指令，每条指令对应着计算机硬件的一个具体操作，利用汇编语言编写的程序与计算机硬件密切相关，程序员可直接对处理器内的寄存器、主存储器的存储单元以及外设的端口等进行操作，从而能够有效地控制硬件。汇编语言程序运行速度快、占用主存容量少，这些都是高级语言无法替代的。所以，作为一名计算机专业人员，必须掌握汇编语言程序设计方法。

1.1 计算机系统概述

计算机系统分为硬件和软件两部分。硬件是计算机系统的机器部分，它是计算机工作的物质基础；软件则是为了运行、管理和维护计算机而编制的各种程序的总和，广义的软件还应该包括与程序有关的文档。利用汇编语言所编写的程序是软件。但是，每条汇编语言指令都使计算机某个具体硬件产生相应的动作，所以，利用汇编语言进行程序设计体现了计算机硬件和软件的结合。

1.1.1 微机系统的基本组成

1. 计算机的硬件

计算机的硬件有五大组成部分：运算器、控制器、存储器、输入设备和输出设备。其中运算器和控制器是计算机的核心，合称中央处理单元(CPU)或处理机、处理器。CPU 内部还有一些高速存储单元，被称为寄存器。输入设备和输出设备往往统称为外部设备，简称为 I/O 设备。在微型计算机中，CPU 由一个大规模集成电路芯片构成，也被称为微处理器。

对汇编语言程序员来说，计算机各部件组成的结构如图 1.1 所示。它们主要由 CPU、存储器和 I/O 设备组成，各部分之间通过总线连接。

CPU 包括运算器、控制器和寄存器组。运算器执行所有的算术和逻辑运算；控制器负责把命令逐条从存储器中取出，经译码分析后向机器发出各种控制命令，如取数、执行、运算和存数，从而正确完成程序所要求的功能。寄存器组为处理单元提供各种操作所需要

的数据。

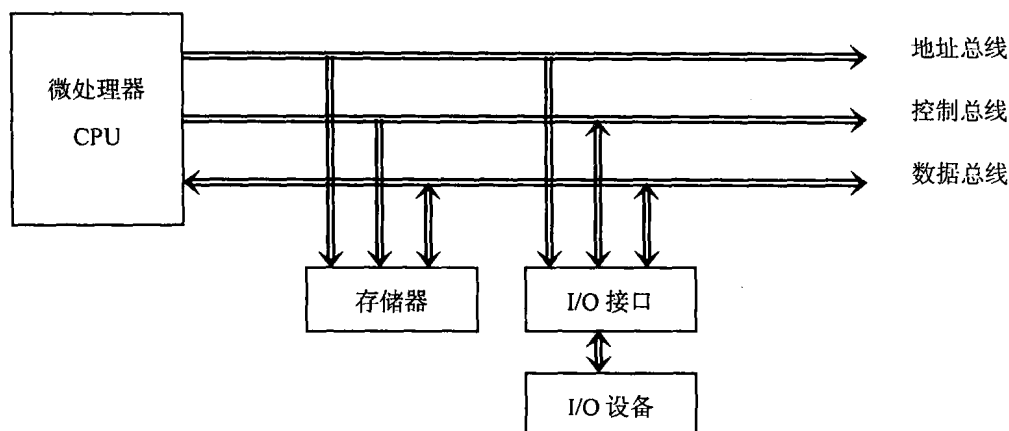


图1.1 计算机结构

存储器是计算机的记忆部件，用来存放程序以及程序中所涉及的数据。存储器的内容不因为被读而消失，它可以重复取出，只有存入新的信息，原来的旧信息才会被更改。

按所在位置，存储器分成主存储器和辅助存储器，分别被简称为主存(内存)和辅存(外存)。主存储器存放当前正在执行的程序和使用的数据，CPU 可以直接存取，它由半导体存储芯片构成，其成本高、容量小，但速度快。辅助存储器可用于长期保存大量程序和数据，它由磁盘或光盘构成，其成本低、容量大，但速度慢。CPU 需要通过 I/O 接口访问。计算机系统中，主存的容量是有限的，需要辅存来补充；辅存的容量比主存大得多，但存取速度却比主存慢得多。一般来说，程序和数据以文件的形式保存在辅存中，只有在用到它们时，才从辅存读到主存的某个区域中，由中央处理单元控制执行。

按读写能力，存储器分为随机存取存储器 RAM 和只读存储器 ROM。CPU 可以从 RAM 取出信息，也可以向 RAM 写入信息，所以 RAM 也被称为读写存储器；而 ROM 中的信息只能被读出，不能被修改。例如磁盘存储器是可读可写的，而 CD-ROM 光盘则是只读的，这两种存储器都是辅存，都可以长期保存数据，但它们的存取速度都慢于半导体存储器。

存储器由大量存储单元组成。为了区别，将每个单元进行编号，每个存储器单元就有一个惟一的存储器地址(Address)。每个存储单元存放 1 个字节的数据，1 个字节 B(Byte)包含 8 个二进制位 b(bit)。

存储容量是指存储器所具有的存储单元个数，其基本单元是字节 B。为了表达更大的容量，常用的单位是 KB(千字节)，MB(兆字节)，GB(吉字节)甚至 TB(太字节)。1 KB=2¹⁰ 字节=1 024 字节，1 MB=2²⁰ 字节，1 GB=2³⁰ 字节，1 TB=2⁴⁰ 字节。

外部设备是实现人机间通信的机电设备。微机系统中，常用的输入设备有键盘、鼠标等；输出设备有显示器、打印机等；起辅存作用的磁盘和光盘也是以外设的形式连接到系统中的，可以认为它们既是输入设备也是输出设备。

由于外设的种类繁多，工作原理各异，所以每个外设必须通过 I/O 接口电路与系统连接。程序员所见的 I/O 接口由一组寄存器组成，为了区别，对各个寄存器进行了编号，就形成了 I/O 地址，通常被称做 I/O 端口(Port)。实际上系统就是通过这些端口与外设进行通信的。

2. 计算机的软件

计算机软件是计算机系统的重要组成部分,分为系统软件 and 用户软件两大类。系统软件是由计算机的生产厂家提供给用户的一组程序,这些程序是用户产生、准备和执行用户程序的基础。用户软件则是用户自行编制的各种程序。

系统软件的核心是操作系统(Operating System)。操作系统是系统程序的集合,它的主要作用是对系统的硬、软件资源进行合理的管理,为用户创造方便、有效和可靠的计算机工作环境。操作系统的主要部分是常驻监督程序(Monitor),只要一开机它就存在于内存中,它可以从用户接受命令,使操作系统执行相应的动作。

I/O 驱动程序用来对 I/O 设备进行控制或管理。当系统程序或用户程序需要使用 I/O 设备时,就调用 I/O 驱动程序对外设发出命令,完成 CPU 和 I/O 设备之间的信息传送。

文件管理(File Management)系统用来处理存储在外存储器中的大量信息,它可以和外存储器的设备驱动程序相连接,对存储在其中的信息以文件(File)形式进行存取、复制及其他操作。

文本是指由字母、数字、符号等组成的信息,它可以是一个用汇编语言或高级语言编写的程序,也可以是一组数据或一份报告。文本编辑(Text Editor)程序用来建立、出入或修改文本,并使它存入存储器或大容量存储器中。例如 IBM PC 机提供的行编程序 Edit 等,可用来建立文件,修改文本,有删除、插入、编辑和显示行等功能;字处理程序 Word 可提供屏幕编辑功能,并能提供各种功能及命令的菜单,使文本的建立和修改更加方便。

我们已经知道计算机是通过逐条地执行组成程序的指令来完成人们所赋予的任务的,所以指令就是计算机能识别并直接执行的语句,它由二进制代码组成,这种语言称为机器语言。

既然计算机能识别的惟一语言是机器语言,用这种语言编写程序很不方便,所以在计算机语言的发展过程中就出现了汇编语言和高级语言。

汇编语言是一种符号语言,它几乎和机器语言一一对应,但在书写时却使用由字符串组成的助记符。例如加法在汇编语言中是用助记符 ADD 表示的,而机器语言则用多位二进制代码表示。显然,相对于机器语言来说,汇编语言是易于为人们所理解的,但计算机却不能直接识别汇编语言。汇编程序就是把用户编制的汇编语言翻译成机器语言的一种系统程序。IBM PC 机中的汇编程序有 ASM 和 MASM 两种。ASM 称为小汇编程序,它占有较小的存储区,但功能较弱;MASM 称为宏汇编程序,它需要的存储区较大,功能较强,具有宏汇编能力。

高级语言脱离机器指令,用更易于理解的方式来编写程序,当然它们也要翻译成机器语言才能在机器上执行。高级语言的翻译有两种方式。一种是先把高级语言程序翻译成机器语言(或先翻译成汇编语言,再由汇编程序翻译成机器语言)程序,然后在机器上执行,这种翻译程序称为编译(Compiler)程序。多数高级语言如 PASCAL 语言、C 语言等都是采用这种方式。另一种是直接把高级语言程序在机器上运行,一边解释一边执行,这种翻译程序称为解释(Interpreter)程序,如 BASIC 语言就采用这种方式。

将高级语言翻译成机器语言的程序称为翻译(Translator)程序。系统程序中的翻译程序包括汇编程序、BASIC 语言解释程序及各种高级语言的编译程序。

连接(Linker)程序用来把要执行的程序与库文件或其他已经翻译的子程序(能完成一种独立功能的程序模块)连接在一起,形成机器能执行的程序。

装入(Loader)程序用来把程序从外存储器传送到内存存储器,以便机器执行。例如,计算机开机后就需要立即启动装入程序以把常驻监督程序装入存储器,使机器运转起来。又如,用户程序经翻译和连接后,由连接程序直接调用装入程序,把可执行的用户程序装入内存以便执行。

调试(Debug)程序是系统提供给用户的能监督和控制用户程序的一种工具。它可以装入、修改、显示或逐条执行一个程序。在 IBM PC 机上,简单的汇编语言程序可以通过 Debug 来建立、修改和执行。

各种标准程序、子程序及一些文件的集合称为程序库,它可以被系统程序或用户程序调用。程序库分为系统程序库(System Library)和用户程序库(User Library)。操作系统还允许用户建立程序库,以提高用户的工作效率。

1.1.2 计算机中数据信息的类型和表示

计算机中的数据信息包括数值数据和非数值数据。数值数据有确定的值,用来表示数量的多少,对它们能进行各种算术运算和处理;其余数据为非数值数据。在非数值数据中有一类最常用的数据,称为字符型数据,它可以方便地表示文字信息。

1. 数制与转换

数制也称为进位计数制。在日常生活中,人们习惯于采用十进制数进行计算,而在计算机中所有的数据都是以二进制代码的形式存储、处理和传送的。但是在输入、输出或书写时,为了用户的方便,也经常用到十六进制。

(1) 十进制数的三个特点。

- ① 由 10 个不同的数码 0、1、2、3、4、5、6、7、8、9 来表示数。
- ② 由低位向高位的进位规则是“逢十进一”。
- ③ 在数中的任何一个数码所表示的数值等于该数码与该位“位权”值的乘积。第 i 位的“位权”值为 10^i 。

十进制数 $a_n a_{n-1} \cdots a_1 a_0 a_{-1} a_{-2} \cdots a_{-m}$ 可以表示为

$$a_n \times 10^n + a_{n-1} \times 10^{n-1} + \cdots + a_1 \times 10^1 + a_0 \times 10^0 + a_{-1} \times 10^{-1} + a_{-2} \times 10^{-2} + \cdots + a_{-m} \times 10^{-m} \cdots$$

(2) 二进制数的三个特点。

- ① 由 2 个不同的数码 0、1 来表示数。
- ② 由低位向高位的进位规则是“逢二进一”。
- ③ 第 i 位的“位权”值为 2^i 。

二进制数 $a_n a_{n-1} \cdots a_1 a_0 a_{-1} a_{-2} \cdots a_{-m}$ 可以表示为

$$a_n \times 2^n + a_{n-1} \times 2^{n-1} + \cdots + a_1 \times 2^1 + a_0 \times 2^0 + a_{-1} \times 2^{-1} + a_{-2} \times 2^{-2} + \cdots + a_{-m} \times 2^{-m} \cdots$$

(3) 十六进制数的 3 个主要特点。

- ① 由 16 个不同的数码 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F 来表示