

高 等 学 校 计 算 机 课 程 规 划 教 材

基于Windows环境的 汇编语言程序设计

钱晓捷 编著



清华大学出版社



高等学校计算机课程规划教材

基于Windows环境的 汇编语言程序设计

钱晓捷 编著

清华大学出版社
北京

内 容 简 介

本书以 32 位英特尔 80x86 处理器和个人计算机为硬件平台,主要介绍基于 Windows 操作系统的 DOS 模拟环境软件平台,借助微软 MASM 汇编程序讲解汇编语言程序设计,内容包括基本的汇编语言基础、常用处理器指令和汇编语言伪指令,以及顺序、分支、循环、子程序结构,最后引申到 32 位 Windows 编程、与 C++ 语言的混合编程,以及浮点、多媒体及 64 位指令等方面。

本书可以作为普通高校“汇编语言程序设计”等课程的教材或参考书,适合计算机及电子、通信和自控等电类专业的本科学士,也适合软件学院、计算机等电类专业的高职、成教学生本科或者专科学生。计算机应用开发人员、希望深入学习汇编语言的普通读者也可以作为入门教材。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

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

基于 Windows 环境的汇编语言程序设计/钱晓捷编著. —北京:清华大学出版社,2012.12
高等学校计算机课程规划教材
ISBN 978-7-302-29885-4

I. ①基… II. ①钱… III. ①汇编语言—程序设计—高等学校—教材 IV. TP313

中国版本图书馆 CIP 数据核字(2012)第 199325 号

责任编辑:汪汉友

封面设计:傅瑞学

责任校对:焦丽丽

责任印制:何 芊

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:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:18.25

字 数:446 千字

版 次:2012 年 12 月第 1 版

印 次:2012 年 12 月第 1 次印刷

印 数:1~3000

定 价:29.50 元

产品编号:048086-01

出版说明

信息时代早已显现其诱人魅力,当前几乎每个人随身都携有多个媒体、信息和通信设备,享受其带来的快乐和便捷。

我国高等教育早已进入大众化教育时代。而且计算机技术发展很快,知识更新速度也在快速增长,社会对计算机专业学生的专业能力要求也在不断翻新。这就使得我国目前的计算机教育面临严峻挑战。我们必须更新教育观念——弱化知识培养目的,强化对学生兴趣的培养,加强培养学生理论学习、快速学习的能力,强调培养学生的实践能力、动手能力、研究能力和创新能力。

教育观念的更新,必然导致教材的更新。一流的计算机人才需要一流的名师指导,而一流的名师需要精品教材的辅助,而精品教材也将有助于催生更多一流名师。名师们在长期的一线教学改革实践中,总结出了一整套面向学生的独特的教法、经验、教学内容等。本套丛书的目的是推广他们的经验,并促使广大教育工作者更新教育观念。

在教育部相关教学指导委员会专家的帮助和指导下,在各大学计算机院系领导的协助下,清华大学出版社规划并出版了本系列教材,以满足计算机课程群建设和课程教学的需要,并将各重点大学的优势专业学科的教育优势充分发挥出来。

本系列教材行文注重趣味性,立足课程改革和教材创新,广纳全国高校计算机专业的一线优秀名师参与,从中精选出佳作予以出版。

本系列教材具有以下特点。

1. 有的放矢

针对计算机专业学生并站在计算机课程群建设、技术市场需求、创新人才培养的高度,规划相关课程群内各门课程的教学关系,以达到教学内容互相衔接、补充、相互贯穿和相互促进的目的。各门课程功能定位明确,并去掉课程中相互重复的部分,使学生既能够掌握这些课程的实质部分,又能节约一些课时,为开设社会需求的新技术课程准备条件。

2. 内容趣味性强

按照教学需求组织教学材料,注重教学内容的趣味性,在培养学习观念、学习兴趣的同时,注重创新教育,加强创新思维、创新能力的培养、训练;强调实践,案例选题注重实际和兴趣度,大部分课程各模块的内容分为基本、加深和拓宽内容3个层次。

3. 名师精品多

广罗名师参与,对于名师精品,予以重点扶持,教辅、教参、教案、PPT、实验大纲和实验指导等配套齐全,资源丰富。同一门课程,不同名师分出多个版本,方便选用。

4. 一线教师亲力

专家咨询指导,一线教师亲力;内容组织以教学需求为线索;注重理论知识学习,注重学

习能力培养,强调案例分析,注重工程技术能力锻炼。

经济要发展,国力要增强,教育必须先行。教育要靠教师和教材,因此建立一支高水平的教材编写队伍是社会发展的需要,特希望有志于教材建设的教师能够加入到本团队。通过本系列教材的辐射,培养一批热心为读者奉献的编写教师团队。

清华大学出版社

前 言

汇编语言是计算机及相关专业的必修课,但是随着技术发展,它已经不适宜作为一个通用程序设计语言那样教学了,而应该更多地用于展示计算机内部的工作原理。基于这样的认识,结合几年的教学实践,我们总结经验编写了本书。

1. 简单易用的开发环境

为了方便教学和初学者入门,本书构建了一个简单易用的开发环境(详见本书第1章),无须安装和配置,直接复制就可使用。它支持16位DOS环境和32位Windows控制台,提供MASM汇编程序、连接程序、CodeView调试程序及其帮助文档,配套输入输出子程序库及方便操作的批处理文件等。另外,第6章和第7章分别引出了MASM 32和Visual C++集成化开发系统,说明使用汇编语言开发32位Windows应用程序和混合编程的方法。

2. 重点明确的教学内容

汇编语言的教学目的是从软件角度理解计算机硬件工作原理,为相关课程提供基础知识,同时全面认识程序设计语言,体会低层编程特点,以便更好地应用高级语言。为此,本书不是详尽展开所有处理器指令、全部汇编伪指令,而是选择处理器通用的基本指令和反映汇编语言特色的常用伪指令;没有引出复杂的程序格式,而是侧重编程思想和技术。这样一方面能够降低教学难度、易于学生掌握,另一方面使得教学内容更加实用,便于学生实际应用。

3. 突出实践的教学过程

本书以约70个例题程序和约60个习题程序贯穿教学内容。第1章在必要的寄存器和存储器知识后,就引出汇编语言开发环境,介绍汇编语言的语句格式、源程序框架和开发方法,并利用简单易用的输入输出子程序编写具有显示结果的程序。第2章结合数据编码,引出汇编语言的常量定义和变量应用,自然掌握常用伪指令。第3章分类学习处理器基本指令,逐渐编写特定要求的程序片段。第4章以程序结构为主线、第5章围绕数码转换子程序、第6章Windows编程、第7章混合编程、第8章浮点指令,从简单到复杂,逐步编写具有实用价值的应用程序。

4. 循序渐进的教学原则

为了便于学生理解和掌握,也便于教师实施教学,本书以“循序渐进、难点分散、前后对照”为原则,做到“语言浅显、描述详尽、图表准确”,内容编排形成诸多特色。例如,将处理器指令和汇编伪指令分散于各个教学内容之中,引出列表文件暂时避开调试程序。程序尽可能具有交互性和趣味性,适当对比高级语言,并展示底层工作原理。每章都编制丰富的习题,满足课外练习、上机任务和试题组织。

为了更好地服务于广大师生和读者,作者维护“大学微机技术系列课程教学辅助网站”(http://www2.zzu.edu.cn/qfw)。网站主要就“汇编语言程序设计”和“微机原理及接口技术”课程提供有关教学课件(电子教案)、教学大纲、教材勘误、疑难解答、输入输出子程序库、示例源程序文件等辅助资源,是本教材的动态延伸。

本书由郑州大学信息工程学院钱晓捷老师主编,并衷心感谢张青、姚俊婷、穆玲玲、关国

利、张行进等老师的协作和帮助。欢迎广大师生和读者通过电子邮件(qianxiaojie@zzu.edu.cn)与作者交流。

作者

2012年9月

教学建议

汇编语言课程主要有周4授课学时(总学时68)和周3授课学时(总学时51)两种教学方案,并配合足够的上机实践学时。结合本书8章内容,各章学时安排参见下表。表中各章主要内容简介用于提示教学重点。

目 录	内 容 简 介	学时(68)	学时(51)
第1章 汇编语言基础	在了解软件开发环境的基础上,熟悉通用寄存器和存储器组织,掌握汇编语言的语句格式、程序框架和开发方法	6	6
第2章 数据表示和寻址	在理解计算机如何表达数据的基础上,熟悉汇编语言中如何使用常量和变量,掌握处理器指令如何寻址数据的方式	8	6
第3章 通用数据处理指令	熟悉 IA-32 处理器数据传送、算术运算、逻辑运算和移位操作等基本指令,通过程序片段掌握指令功能和编程应用	10	8
第4章 程序结构	以顺序、分支和循环程序结构为主线,结合数值运算、数组处理等示例程序,掌握控制转移指令以及编写基本程序的方法	14	12
第5章 模块化程序设计	以子程序结构为主体,围绕数码转换实现键盘输入和显示输出,掌握子程序、文件包含、宏汇编等各种多模块编程的方法	12	8
第6章 Windows 编程	熟悉汇编语言调用 API 函数的方法,掌握控制台输入输出函数。熟悉 MASM 的高级特性,理解 Windows 图形窗口程序的编写	10	6
第7章 与 Visual C++ 的混合编程	掌握嵌入汇编和模块连接进行混合编程方法,理解堆栈帧的作用,熟悉汇编语言调用高级语言函数和开发调试过程	6	4
第8章 浮点、多媒体及 64 位指令	熟悉浮点数据格式、多媒体数据格式及 64 位编程环境的特点,了解浮点操作、多媒体操作和 64 位指令	2	1

目 录

第 1 章 汇编语言基础	1
1.1 英特尔 80x86 处理器	1
1.1.1 16 位 80x86 处理器	1
1.1.2 IA-32 处理器	2
1.1.3 Intel 64 处理器	3
1.2 个人计算机系统	4
1.2.1 硬件组成.....	4
1.2.2 寄存器.....	6
1.2.3 存储器组织	10
1.2.4 程序设计语言	16
1.2.5 软件系统	18
1.3 汇编语言程序格式.....	21
1.3.1 指令代码格式	21
1.3.2 语句格式	24
1.3.3 源程序格式	26
1.3.4 开发过程	32
习题 1	37
第 2 章 数据表示和寻址	39
2.1 数据表示.....	39
2.1.1 数制	39
2.1.2 数值的编码	42
2.1.3 字符的编码	44
2.2 常量表达.....	47
2.3 变量应用.....	49
2.3.1 变量定义	50
2.3.2 变量属性	55
2.4 数据寻址方式.....	58
2.4.1 立即数寻址	58
2.4.2 寄存器寻址	59
2.4.3 存储器寻址	60
2.4.4 数据寻址的组合	65

习题 2	66
第 3 章 通用数据处理指令	70
3.1 数据传送类指令	70
3.1.1 通用传送指令	70
3.1.2 堆栈操作指令	72
3.1.3 其他传送指令	75
3.2 算术运算类指令	80
3.2.1 状态标志	80
3.2.2 加法指令	82
3.2.3 减法指令	84
3.2.4 乘法和除法指令	85
3.2.5 其他运算指令	87
3.3 位操作类指令	89
3.3.1 逻辑运算指令	89
3.3.2 移位指令	93
习题 3	97
第 4 章 程序结构	104
4.1 顺序程序结构	104
4.2 分支程序结构	105
4.2.1 无条件转移指令	105
4.2.2 条件转移指令	109
4.2.3 单分支结构	113
4.2.4 双分支结构	114
4.2.5 多分支结构	116
4.3 循环程序结构	119
4.3.1 循环指令	119
4.3.2 计数控制循环	121
4.3.3 条件控制循环	122
4.3.4 多重循环	124
4.3.5 串操作指令	126
习题 4	131
第 5 章 模块化程序设计	136
5.1 子程序结构	136
5.1.1 子程序指令	136
5.1.2 子程序设计	139
5.2 参数传递	140

5.2.1	寄存器传递参数	141
5.2.2	共享变量传递参数	144
5.2.3	堆栈传递参数	147
5.3	多模块程序结构	150
5.3.1	源文件包含	150
5.3.2	模块连接	154
5.3.3	子程序库	155
5.4	宏结构	156
5.4.1	宏汇编	156
5.4.2	重复汇编	163
5.4.3	条件汇编	165
	习题 5	168
第 6 章	Windows 编程	172
6.1	操作系统函数调用	172
6.1.1	动态链接库	172
6.1.2	MASM 的过程声明和调用	173
6.1.3	程序退出函数	174
6.1.4	Windows 程序格式	175
6.2	控制台应用程序	176
6.2.1	控制台输出	176
6.2.2	控制台输入	179
6.3	图形窗口应用程序	182
6.3.1	消息窗口	182
6.3.2	结构变量	184
6.3.3	MASM 的高级语言特性	186
6.3.4	简单窗口程序	194
	习题 6	201
第 7 章	与 Visual C++ 的混合编程	205
7.1	嵌入汇编	205
7.2	模块连接	209
7.2.1	约定规则	209
7.2.2	堆栈帧	211
7.3	调用高级语言函数	219
7.3.1	嵌入汇编中调用高级语言函数	219
7.3.2	汇编语言中调用 C 库函数	220
7.4	使用 Visual C++ 开发环境	221
7.4.1	汇编语言程序的开发过程	221

7.4.2 汇编语言程序的调试过程.....	223
习题 7	226
第 8 章 浮点、多媒体及 64 位指令	231
8.1 浮点指令	231
8.1.1 实数编码.....	232
8.1.2 浮点寄存器.....	235
8.1.3 浮点指令编程.....	238
8.2 多媒体指令	241
8.2.1 MMX	242
8.2.2 SSE	244
8.2.3 SSE2	246
8.2.4 SSE3	247
8.3 64 位指令	248
8.3.1 64 位方式的运行环境	249
8.3.2 64 位方式的指令	250
习题 8	251
附录 A 调试程序 CodeView	254
A.1 CodeView 的菜单命令	255
A.2 CodeView 的窗口	256
A.3 CodeView 的设置	258
A.4 使用 CodeView 的调试示例	260
附录 B 常用 DOS 功能调用	266
附录 C 输入输出子程序库	267
附录 D 32 位通用指令列表	270
附录 E MASM 伪指令和操作符列表	275
附录 F 列表文件符号说明	277
附录 G 常见汇编错误信息	278
参考文献	281

第 1 章 汇编语言基础

程序设计语言是人与计算机沟通的语言,程序员利用它进行软件开发。通常人们习惯使用类似自然语言的高级程序设计语言,例如 C、C++ 或者 BASIC、Java 语言等。高级语言需要翻译为计算机能够识别的指令,即机器语言,才能被计算机直接执行。机器语言是一串 0 和 1 组成的二进制代码,对程序员来说艰涩难懂,被称为低级语言。将二进制代码的指令和数据用便于记忆的符号,即助记符(Mnemonic)表示就形成汇编语言(Assembly),所以汇编语言是一种面向机器的低级程序设计语言,或称为低层语言。

本章首先介绍汇编语言的硬件基础:英特尔 80x86 处理器和个人计算机,然后是软件基础: DOS、Windows 操作系统和微软 MASM 汇编程序,接着理解汇编语言的意义,最后学习汇编语言的程序格式,并编写第一个汇编语言程序。

1.1 英特尔 80x86 处理器

汇编语言的主体是处理器指令。处理器(P processor)是计算机的运算和控制核心,也常被称为中央处理单元(Central Processing Unit, CPU)。微型计算机中的处理器常采用一块大规模集成电路芯片构成,也就被称为微处理器(Microprocessor),它代表着整个微型计算机系统的性能。所以,通常就将采用微处理器为核心构造的计算机称为微型计算机。

微型计算机(Microcomputer)是人们最常使用的一类计算机,在科学计算、信息管理、自动控制、人工智能等领域有着广泛的应用。工作、学习和娱乐中使用的个人计算机(PC)是人们最熟悉,也是最典型的通用微型计算机。

美国英特尔(Intel)公司是目前世界上最有影响的处理器生产厂家,也是世界上第一个微处理器芯片的生产厂家。英特尔 80x86 系列处理器一直是个人计算机的主流处理器。

1.1.1 16 位 80x86 处理器

1971 年,英特尔公司生产的 4 位处理器芯片 Intel 4004 宣告了微型计算机时代的到来。1972 年,英特尔公司开发了 8 位处理器 Intel 8008 芯片;1974 年接着生产了 Intel 8080;1977 年,英特尔公司将 8080 及其支持电路集成在一块集成电路芯片上,形成了性能更高的 8 位处理器 8085。1978 年,英特尔公司在其 8 位处理器基础上,陆续推出了 16 位结构的 8086、8088 和 80286 等处理器;它们在 IBM PC 系列机中获得广泛应用,被称为 16 位 80x86 处理器。

1. 8086

1978 年,英特尔推出 16 位 8086 处理器,这是该公司生产的第一个 16 位结构处理器芯片。8086 芯片的对外引脚共有 40 个,其中包括 16 位数据总线,20 位地址总线,支持 1MB (1 兆字节)主存容量、5MHz (5 兆赫兹)时钟频率。8086 具有的所有指令,即指令系统(Instruction Set)成为整个 Intel 80x86 系列处理器的 16 位基本指令集。

为了方便与当时的 8 位外部设备连接,1979 年,英特尔推出了被称为“准 16 位处理器”的 Intel 8088。8088 只是将外部数据总线设计为 8 位,内部仍保持 16 位结构,指令系统等都与 8086 相同。随后的 Intel 80186 和 Intel 80188 则是分别以 8086 和 8088 为核心,并配以支持电路构成的芯片;但它们在 8086 指令系统的基础上新增若干条实用指令,涉及堆栈操作、输入输出指令、移位指令、乘法指令、支持高级语言的指令。

2. 80286

1982 年,英特尔推出仍为 16 位结构的 Intel 80286 处理器,但地址总线扩展为 24 位,即主存储器具有 16MB 容量。Intel 80286 设计有与 8086 工作方式一样的实方式(Real Mode),还新增有保护方式(Protected Mode)。在实方式下,Intel 80286 相当于一个快速 8086。在保护方式下,Intel 80286 提供了存储管理、保护机制和多任务管理的硬件支持。为支持保护方式,80286 引入了系统指令,为操作系统等核心程序提供处理器控制功能。

1.1.2 IA-32 处理器

IBM PC 系列机的广泛应用推动了处理器芯片的生产。英特尔公司在推出 32 位结构的 80386 处理器后,明确宣布 Intel 80386 芯片的指令集结构 ISA (Instruction Set Architecture)被确定为以后开发的 80x86 系列处理器的标准,称为英特尔 32 位结构:IA-32(Intel Architecture-32)。现在,英特尔公司的 80386、80486 以及 Pentium 各代处理器被通称为 IA-32 处理器或 32 位 80x86 处理器。

1. 80386

1985 年,Intel 80x86 处理器进入第 3 代 80386。Intel 80386 处理器采用 32 位结构,数据总线 32 位,地址总线也是 32 位,可寻址 4GB(吉字节)主存,时钟频率有 16MHz、25MHz 和 33MHz。IA-32 指令系统在兼容原 16 位 80286 指令系统基础上,全面升级为 32 位,还新增了有关位操作、条件设置等指令。

Intel 80386 除保持与 Intel 80286 兼容外,又提供了虚拟 8086 工作方式(Virtual 8086 Mode)。虚拟 8086 方式是在保护方式下的一种特殊状态,类似 8086 工作方式但又接受保护方式的管理,能够模拟多个 8086 处理器。32 位 PC 的 Windows 操作系统采用保护方式,其 MS-DOS 命令行(环境)就是虚拟 8086 方式,而早期采用的 DOS 操作系统是以实方式为基础建立的。

2. 80486

1989 年,英特尔公司出品 80486 处理器。它内部集成了 120 万个晶体管,最初的时钟频率为 25MHz,很快发展到 33MHz 和 50MHz。从结构上来说,Intel 80486 把 Intel 80386 处理器与 80387 数学协处理器和 8KB 高速缓冲存储器(Cache)集成在一个芯片上,使处理器的性能大大提高。

传统上,中央处理器 CPU 主要是整数处理器。为了协助处理器处理浮点数据(实数),英特尔设计有数学协处理器,后被称为浮点处理单元(Floating-Point Unit, FPU)。配合 8086 和 8088 整数处理器的数学协处理器是 8087,配合 80286 的是 80287,80386 采用 80387。而从 Intel 80486 开始,FPU 已经被集成到一个处理器当中,IA-32 处理器的指令系统也就包含了浮点指令,能够直接支持对浮点数据的处理。

3. Pentium 系列

Pentium 芯片原来应该被称为 80586 处理器,因为数字很难进行商标版权保护的缘故而特意取名。其实,Pentium 源于希腊文 pente(数字 5),加上后缀-ium(化学元素周期表中命名元素常用的后缀)变化而来的。同时,英特尔公司为其取了一个响亮的中文名称:奔腾,并进行了商标注册,形成了系列产品。

英特尔公司于 1993 年制造成功 Pentium,于 1995 年正式推出 Pentium Pro(原来被称为 P6,中文名称为“高能奔腾”)。在处理器结构上,Pentium 主要引入了超标量(Superscalar)技术,Pentium Pro 主要采用了动态执行技术来提升处理器性能。它们增加了若干整数指令,完善了浮点指令。

前面所述的各代 IA-32 处理器,都新增有若干实用指令,但非常有限。为了顺应微型计算机向多媒体和通信方向发展,英特尔公司及时在其处理器中加入了多媒体扩展 MMX (MutliMedia eXtension)技术。MMX 技术于 1996 年正式公布,它在 IA-32 指令系统中新增了 57 条整数运算多媒体指令,可以用这些指令对图像、音频、视频和通信方面的程序进行优化,使微型机对多媒体的处理能力较原来有了大幅度提升。MMX 指令应用于 Pentium 处理器就是 Pentium MMX(多能奔腾)。MMX 指令应用于 Pentium Pro 处理器就是 Pentium II,它于 1997 年推出。

1999 年,针对国际互联网和三维多媒体程序的应用要求,英特尔在 Pentium II 的基础上又新增了 70 条 SSE(Streaming SIMD Extensions)指令(原称为 MMX-2 指令),开发了 Pentium III。SSE 指令侧重于浮点单精度多媒体运算,极大地提高了浮点 3D 数据的处理能力。SSE 指令类似于 AMD 公司发布的 3D Now! 指令。由于这些多媒体指令具有显著的单指令多数据(Single Instruction Multiple Data, SIMD)处理能力,即一条指令可以同时进行了多组数据的操作,现在统称为 SIMD 指令。

2000 年 11 月,英特尔公司推出 Pentium 4,新增 76 条 SSE2 指令集,侧重于增强浮点双精度多媒体运算能力。2003 年的新一代 Pentium 4 处理器,又新增了 13 条 SSE3 指令,用于补充完善 SIMD 指令集。

1.1.3 Intel 64 处理器

随着互联网、多媒体、3D 视频等的发展,信息时代的应用对计算机性能提出了越来越高的要求,32 位单核处理器已不能适应这一要求。

1. Intel 64 结构

一直以来,80x86 处理器的更新换代都保持与早期处理器的兼容,以便继续使用现有的软硬件资源。但是,英特尔公司迟迟不愿将 80x86 处理器扩展为 64 位,这给了超威(AMD)公司一个机会。AMD 公司是生产 IA-32 处理器兼容芯片的厂商,是英特尔公司最主要的竞争对手。AMD 公司的 IA-32 兼容处理器,其价格低于英特尔芯片,但性能却没有超越对应的英特尔芯片。于是,AMD 公司于 2003 年 9 月率先推出支持 64 位、兼容 80x86 指令集结构的 Athlon 64 处理器(K8 核心),将桌面 PC 引入了 64 位领域。

2005 年,在 PC 用户对 64 位技术的企盼和 AMD 公司 64 位处理器的压力下,英特尔公司推出了扩展存储器 64 位技术(Intel EM64T: Intel Extended Memory 64 Technology)。EM64T 技术是 IA-32 结构的 64 位扩展,首先应用于支持超线程技术的 Pentium 4 终极版

(支持双核技术)和 6xx 系列 Pentium 4 处理器。随着 EM64T 技术的出现,IA-32 指令系统也扩展成为 64 位,后来被称为 Intel 64 结构。这之后的 Pentium 4 处理器,以及 Pentium E 系列多核处理器、酷睿(Core)2 和酷睿 i 系列多核处理器等都支持 Intel 64 结构。

Intel 64 位结构为软件提供了 64 位线性地址空间,支持 40 位物理地址空间。IA-32 处理器支持保护方式(含虚拟 8086 方式)、实地址方式和系统管理 SMM 方式,Intel 64 位结构则引入了一个新的工作方式:32 位扩展工作方式(IA-32e)。IA-32e 除了有一个运行 32 位和 16 位软件的兼容方式,还有一个 64 位方式。在 64 位工作方式,允许 64 位操作系统运行存取 64 位地址空间的应用程序,还可以存取 8 个附加的通用寄存器、8 个附加的 SIMD 多媒体寄存器、64 位通用寄存器和 64 位指令指针等。

2. 多核技术

单纯以提高时钟频率等传统的增加处理器复杂度的方法已经很难提升处理器性能,还带来功耗剧增、发热量巨大的问题。于是,多核(Multi-core)技术应运而生。多核处理器是在一个集成电路芯片上制作了两个或多个处理器执行核心,依靠多个处理器核心相互协作同时执行多个程序线程提升性能。基于不同的处理器内部结构,英特尔也推出多款多核处理器,目前主要是英特尔奔腾 E 系列多核处理器、酷睿 2 和酷睿 i 系列多核处理器。

另外,SSE 系列指令集继续丰富,酷睿 2 补充了 SSE3 指令(即 32 条 SSSE3 指令),又推出增加了 54 条指令的 SSE4 指令集。其中 47 条指令在英特尔面向服务器领域的至强(Xeon)5400 系列和酷睿 2 至尊版 QX 9650 引入,被称为 SSE4.1 指令,致力于提升多媒体、3D 处理等的性能;其余 7 条指令被称为 SSE4.2 指令。

英特尔公司充分利用集成电路生产的先进技术和处理器结构的革新技术,推出了多种英特尔 80x86 系列处理器芯片。就目前的发展来看,英特尔公司正在利用单芯片多处理器技术生产双核、4 核等多核处理器,并推广支持 64 位处理器和 64 位软件的个人计算机。

1.2 个人计算机系统

计算机系统包括硬件和软件两大部分。硬件(Hardware)是指构成计算机的实在的物理设备,是人们看得见、摸得着的物体,就像人的躯体。软件(Software)一般是指在计算机上运行的程序(广义的软件还包括由计算机管理的数据以及有关的文档资料),是人们指示计算机工作的命令,就像人的思想。

1.2.1 硬件组成

源于冯·诺依曼设计思想的计算机由五大部件组成:控制器、运算器、存储器、输入设备和输出设备。控制器是整个计算机的控制核心;运算器是对信息进行运算处理的部件;存储器是用来存放数据和程序的部件;输入设备将数据和程序变换成计算机内部所能识别和接受的信息方式,并把它们送入存储器中;输出设备将计算机处理的结果以人们能接受的其他机器能接受的形式送出。

现代计算机在很多方面都对冯·诺依曼计算机结构进行了改进,例如五大部件体现在现代计算机中,则是 3 个硬件子系统:处理器、存储系统和输入输出系统。处理器(中央处理单元 CPU)包括运算器和控制器,是信息处理的中心部件,现在都被制作在一起、形成处

处理器芯片。存储系统由寄存器、高速缓冲存储器、主存储器和辅助存储器构成层次结构。处理器和存储系统在信息处理中起主要作用,是计算机硬件的主体部分,通常被称为“主机”。输入(Input)设备和输出(Output)设备统称为外部设备,简称为外设或 I/O 设备;输入输出系统的主体是外设,但还包括外设与主机之间相互连接的 I/O 接口电路。

为简化各个部件的相互连接,现代计算机广泛应用总线结构,如图 1-1 所示。采用总线连接系统中各个功能部件使得计算机系统具有了组合灵活、扩展方便的特点。

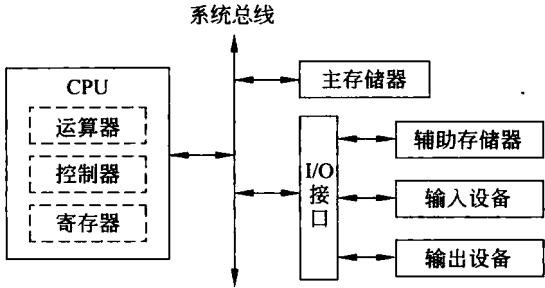


图 1-1 计算机系统的硬件组成

1. 处理器

计算机的核心是处理器(CPU),微型计算机也常称为微处理器。它是采用大规模集成电路技术生产的半导体芯片,芯片内集成了控制器、运算器和若干高速存储单元(即寄存器)。高性能处理器内部非常复杂,例如运算器中不仅有基本的整数运算器,还有浮点处理单元甚至多媒体数据运算单元,控制器还会包括存储管理单元、代码保护机制等。处理器及其支持电路构成了计算机系统的处理和控制中心,对系统的各个部件进行统一的协调和控制。

16 位 IBM PC 系列机采用 16 位 80x86 处理器,32 位 PC 系列机则采用 IA-32 处理器或其兼容芯片(例如 AMD 公司的系列处理器)。

2. 存储器

存储器(Memory)是存放程序和数据 的部件。存储系统由处理器内部的寄存器(Register)、高速缓冲存储器,主板上的主存储器和以外设形式出现的辅助存储器构成。

主存储器(简称主存或内存)由半导体存储器芯片组成,安装在计算机内部的电路板上,相对辅助存储器来说造价高、速度快,但容量小,主要用来存放当前正在运行的程序和正待处理的数据。辅助存储器(简称辅存或外存)主要由磁盘、光盘存储器等构成,以外设的形式安装在计算机上,相对主存储器造价低、容量大、信息可长期保存,但速度慢,主要用来长久保存程序和数据。

从读写功能区别半导体存储器,它被分为可读可写的随机存取存储器(Random Access Memory, RAM)和只读存储器(Read Only Memory, ROM)。构成主存既需要 RAM,也需要 ROM,但注意半导体 RAM 芯片在断电后原存放信息将会丢失,而 ROM 芯片中的信息可在断电后长期保存。

个人计算机主存由半导体存储芯片 ROM 和 RAM 构成。ROM 部分主要是固化的 ROM-BIOS。BIOS(Basic Input/Output System,基本输入输出系统)是 PC 软件系统最底层的程序。它由诸多子程序组成,主要用来驱动和管理诸如键盘、显示器、打印机、磁盘、时

钟、串行通信接口等基本的输入输出设备。操作系统通过对 BIOS 的调用驱动各硬件设备，用户也可以在应用程序中调用 BIOS 中的许多功能。

ROM 空间还包含机器复位后初始化系统的程序，它将操作系统引导到 RAM 空间执行。在 16 位 PC 系列机时代，RAM 容量不过是 64KB 或 1MB。32 位 PC 的 RAM 容量从最初的 4MB，逐渐发展直到 2010 年的 2GB 或 4GB。由于大量应用程序都需要 RAM 主存空间，因此 PC 的主存主要由 RAM 构成，俗称主存条(内存条)。

3. 外部设备

外部设备是指计算机上配备的输入设备和输出设备，也称 I/O 设备或外围设备(简称外设, Peripheral)，其作用是让用户与计算机实现交互。

个人计算机上配置的标准输入设备是键盘、标准输出设备是显示器，二者又合称为控制台(Console)。个人计算机还可用鼠标器、打印机等 I/O 设备。作为外部存储器驱动装置的磁盘驱动器，既是输出设备，又是输入设备。

由于各种外设的工作速度、驱动方法差别很大，无法与处理器直接匹配，因此不可能将它们直接连接到主机。这里就需要有一个 I/O 接口来充当外设和主机间的桥梁，通过该接口电路来完成信号变换、数据缓冲、联络控制等工作。在个人计算机中，较复杂的 I/O 接口电路常制成独立的电路板(也常被称为接口卡 Card)，使用时将其插在主板上。

4. 系统总线

总线(Bus)是用于多个部件相互连接、传递信息的公共通道，物理上就是一组公用导线。例如，处理器芯片的对外引脚(Pin)常被称为处理器总线。这里的系统总线(System Bus)是指计算机系统中主要的总线，例如处理器与存储器和 I/O 设备进行信息交换的公共通道。

16 位 PC 采用 16 位工业标准结构(Industry Standard Architecture, ISA)系统总线连接各个功能部件。32 位 PC 上使用 PCI(Peripheral Component Interconnect, 外设部件互连)总线连接 I/O 接口卡。系统总线除了作为主机板上处理器、主存和 I/O 接口的公共通道外，主板上还设置有许多系统总线插槽，主要用于插接 I/O 接口电路以扩充系统连接的外设，故也被称为 I/O 通道。

对汇编语言程序员来说，处理器、存储器和外部设备依次被抽象为寄存器、存储器地址和输入输出地址，因为编程过程中只能通过寄存器和地址实现处理器控制、存储器和外部设备的数据存取和处理等操作。下面具体学习 IA-32 处理器的寄存器和存储器组织，而输入输出地址将在本书第 3.1 节介绍。

1.2.2 寄存器

处理器内部需要高速存储单元，用于暂时存放程序执行过程中的代码和数据，这些存储单元被称为寄存器(Register)。处理器内部设计有多种寄存器，每种寄存器还可能有多，从应用的角度可以分成两类：透明寄存器和可编程寄存器。

有些寄存器对应用人员来说不能通过指令直接编程控制，例如用于保存指令代码的指令寄存器。它们对应用人员来说好像看不见一样，被称为透明寄存器。这里的“透明”(Transparency)是计算机学科中常用的一个专业术语，表示实际存在但从某个角度看好像没有；运用“透明”思想可以使我们抛开不必要的细节，而专注于关键问题。