21世纪高等学校计算机学科系列教材

汇编语言程序设计

徐建民　等编著
王凤先　审

全国高等学校计算机教育研究会
课程与教材建设委员会推荐出版

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: http://www.phei.com.cn

内 容 简 介

本书以 80x86/Pentium 系列微处理器为背景，系统地介绍了汇编语言程序设计的基础知识、程序设计方法和应用技术。

全书内容共 10 章，可以分为三部分：第 1 章～第 4 章为第一部分，是整个汇编语言程序设计的基础，分别介绍了数在计算机中的表示、汇编语言运行的硬件环境。第 5、6、7、8 章为第二部分，是本书的核心，详细介绍了 80x86 和 Pentium 微型计算机汇编语言程序设计的基本方法和技巧。第 9、第 10 两章为第三部分，分别介绍了汇编语言和高级语言的混合编程和保护模式下汇编语言程序设计的基本方法和应用技术。

本书可以作为计算机科学技术和相关专业本、专科教材，也可以作为从事相关技术工作人员的参考书。

图书在版编目(CIP)数据

汇编语言程序设计/徐建民等编著 . - 北京：电子工业出版社，2001.9
21 世纪高等学校计算机学科系列教材
ISBN 7-5053-6706-4

Ⅰ. 汇… Ⅱ. 徐… Ⅲ. 汇编语言 - 程序设计 - 教材 Ⅳ. TP313

中国版本图书馆 CIP 数据核字(2001)第 057204 号

丛 书 名：21 世纪高等学校计算机学科系列教材
书　　名：**汇编语言程序设计**
编 著 者：徐建民　等
审　　者：王凤先
责任编辑：吕　迈
特约编辑：吴浩源
排版制作：电子工业出版社计算机排版室
印 刷 者：北京东光印刷厂
装 订 者：三河市万和装订厂
出版发行：电子工业出版社　URL:http://www.phei.com.cn
　　北京市海淀区万寿路 173 信箱　邮编 100036
经　　销：各地新华书店
开　　本：787×1092　1/16　印张：19.5 字数：500 千字
版　　次：2001 年 9 月第 1 版　2001 年 9 月第 1 次印刷
书　　号：ISBN 7-5053-6706-4/TP·3745
印　　数：10 100 册　　定价：22.00 元

序　言

这套教材是面向21世纪计算机学科系列教材。为什么要组织这套教材？根据什么编写这套教材？这些都是在这篇序言中要回答的问题。

计算机学科是一个飞速发展的学科，尤其是近十年来，计算机向高度集成化、网络化和多媒体化发展的速度一日千里。但是，从另一个方面来看，目前高等学校的计算机教育，特别是教材建设，远远落后于现实的需要。现在的教材主要是根据《教学计划1993》的要求组织编写的。这个教学计划，在制定过程中主要参照了美国IEEE和ACM的《教学计划1991》。

10年来，计算机学科已有了长足发展，这就要求高等学校计算机教育必须跟上形势发展的需要，在课程设置和教材建设上做出相应调整，以适应面向21世纪计算机教育的要求。这是组织这套教材的初衷。

为了组织好这套教材，全国高等学校计算机教育研究会课程与教材建设委员会在天津召开了“全国高等学校计算机学科课程与教材建设研讨会”，在北京召开了“教材编写大纲研讨会”。在这两次会议上，代表们深入地研讨了全国高校计算机专业教学指导委员会和中国计算机学会教育委员会制定的《计算机学科教学计划2000》以及美国IEEE和ACM的《计算机学科教学计划2001》，这是这套教材参照的主要依据。

IEEE和ACM的《计算机学科教学计划2001》是在总结了从《计算机学科教学计划1991》到现在，计算机学科十年来发展的主要成果的基础上诞生的。它认为面向21世纪计算机学科应包括14个主科目，其中12个主科目为核心主科，它们是：算法与分析(AL)、体系结构(AR)、离散结构(DS)、计算科学(CN)、图形学、可视化、多媒体(GR)、网络计算(NC)、人机交互(HC)、信息管理(IM)、智能系统(IS)、操作系统(OS)、程序设计基础(PF)、程序设计语言(PL)、软件工程(SE)、社会、道德、法律和专业问题(SP)。其中除CN和GR为非核心主科目外，其他12项均为核心主科目。

将2001教学计划与1991教学计划比较可看出：

(1) 在1991年计划中，离散结构只作为数学基础提出，而在2001计划中，则作为核心主科目提出，显然，提高了它在计算机学科中的地位。

(2) 在1991计划中，未提及网络计算，而在2001计划中，则作为核心主科目提出，以适应网络技术飞速发展的需求。

(3) 图形学、可视化与多媒体也是为适应发展要求新增加的内容。

除此之外，2001计划在下述5个方面做调整：

将程序设计语言引论调整为程序设计基础，将人－机通信调整为人机交互，将人工智能与机器人学调整为智能系统，将数据库与信息检索调整为信息管理，将数值与符号计算调整为计算科学。

显然，这些变化使2001计划更具有科学性，也更好地适应了学科发展的需要。

在组织这套教材的过程中，充分考虑了这些变化和调整，在软件和硬件的课程体系、界面划分方面均做了相应的调整，使整套教材更具有科学性和实用性。

另外，还要说明一点，教材建设既要满足必修课的要求，又要满足限选课和任选课的要求。

因此，教材应按系列组织，反映整个计算机学科的要求，采用大拼盘结构，以适应各校不同的具体教学计划，使学校可根据自己的需求进行选择。

这套教材包括：《微机应用基础》、《离散数学》、《电路与电子技术》、《电路与电子技术习题与实验指南》、《数字逻辑与数字系统》、《计算机组成原理》、《微机接口技术》、《计算机体系结构》、《计算机网络》、《计算机网络实验教程》、《通信原理》、《计算机网络管理》、《网络信息系统集成》、《多媒体技术》、《计算机图形学》、《计算机维护技术》、《数据结构》、《计算机算法设计与分析》、《计算机数值分析》、《汇编语言程序设计》、《Pascal 语言程序设计》、《VB 程序设计》、《C 语言程序设计》、《C + + 语言程序设计》、《Java 语言程序设计》、《操作系统原理》、《UNIX 操作系统原理与应用》、《Linux 操作系统》、《软件工程》、《数据库系统原理》、《编译原理》、《编译方法》、《人工智能》、《计算机信息安全》、《计算机图像处理》、《人机交互》、《计算机伦理学》。对于 IEEE 和 ACM 的《计算机学科教学计划 2001》中提出的 14 个主科目，这套系列教材均涵盖，能够满足不同层次院校、不同教学计划的要求。

这套系列教材由全国高等学校计算机教育研究会课程与教材建设委员会主任李大友教授精心策划和组织。编者均为具有丰富教学实践经验的专家和教授。所编教材体系结构严谨、层次清晰、概念准确、论理充分、理论联系实际、深入浅出、通俗易懂。

教材组织过程中，得到了哈尔滨工业大学蒋宗礼教授，西安交通大学董渭清副教授，武汉大学张焕国教授，吉林大学张长海教授，福州大学王晓东教授，太原理工大学余雪丽教授等的大力支持和帮助，在此一并表示衷心感谢。

李大友

2000 年 6 月

前　言

汇编语言程序设计是计算机科学与技术专业的一门重要课程，也是其他相关专业的一门必修或选修课。利用汇编语言可以编写出时空效率高的程序，在某些领域，汇编语言仍然是必不可少的编程语言之一。

目前，整个中国高等教育正处于教学改革的年代，新的教学思路、新的课程体系和教学内容正在形成。根据全国高等学校计算机教育研究会课程与教材建设委员会的指导意见，结合面向21世纪计算机科学与技术专业课程改革的基本思路，在几年教学实践的基础上，我们编写了这本《汇编语言程序设计》。

考虑到国内广泛使用的微型计算机都是以Intel的80x86/Pentium系列微处理器或者兼容的微处理器为CPU的，所以本书以80x86/Pentium系列微处理器为基础，系统地介绍汇编语言程序设计的基础知识、程序设计方法和应用技术。

全书共分10章。前两章概括地介绍数在计算机中的表示，汇编语言运行的硬件环境——80x86和Pentium微处理器以及存储器的组成和结构，它是后续内容的基础。对于已经了解微型计算机原理的读者，这部分内容可以跳过。第3章、第4章介绍寻址方式、指令系统和汇编语言的程序结构。第5、6、7、8章是本书的核心部分，详细地介绍了80x86和Pentium微型计算机汇编语言程序设计的基本方法和技巧。第9章介绍汇编语言和高级语言的混合编程方法。第10章概要介绍保护模式下汇编语言程序设计的基本方法和应用技术。考虑到新教学计划的课时安排，本书语言尽可能简练，故关于浮点数编程的内容未包括在内。

本书第1章至第3章、第7章、第8章由徐建民编写，第4章至第6章由袁方编写，第9章和第10章由杨晓辉编写。全书由徐建民统稿。

本书的编写得到北京工业大学李大友老师的大力支持，王凤先教授在百忙中审阅了全书并且提出了宝贵的修改意见，在此对他们的帮助表示衷心的感谢。本书的初稿曾在河北大学试用，得到了有关老师和同学的许多帮助，在此一并表示感谢。

由于编者能力所限，加之编写时间仓促，书中不妥甚至错误在所难免，恳切希望读者批评指正。

作　者

2001年8月

目　录

第1章 基 础 知 识

1.1 数据表示方法

1.1.1 数与数制

数可以用不同的计数制来表示，习惯上常用十进制计数法。在计算机中，为了便于信息存储和计算，采用二进制计数法。一般地，基数为 r 的 r 进制数的值可以表示为：

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

其中 a_i 可以是 0，1，…，r−1 中的任一数码，r^i 则是各位的权。

1. 二进制数(Binary)

二进制数由 0、1 两个数码构成，基数为 2，第 i 位的权为 2^i，运算逢二进一。为了和十进制数区别，书写时在尾部加注字母 B。

二进制数 $a_n a_{n-1} \cdots b_{-1} b_{-2} \cdots b_{-m}$ 的值为：

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

其中 a_i 和 b_i 为 0，1 两个数码中一个。

例如：$101011B = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 43_{10}$

n 位二进制数可以表示 2^n 种组合。3 位能够表示 8 种组合，0 至 7 的整数；4 位能够表示 16 种组合，0 至 15 的整数；8 位能够表示 256 种组合，0 至 255 的整数，16 位能表示 64×1024(64K)种组合。

2. 八进制数(Octal)和十六进制数(Hexadecimal)

二进制数在计算机中容易实现，易于存储，抗干扰性强；但二进制数的一个很大缺点是表示一个数所需位数多，人们阅读、书写、记忆等不太方便。例如十进制数 1000_{10}，用二进制数表示则需要 10 位二进制数字 1111101000_2。为了便于人们阅读与书写，经常用八进制数或十六进制数来代替二进制数。

八进制数由 0，1，2，3，4，5，6，7 八个数码构成，基数为 8，第 i 位的权为 8^i，运算逢八进一。

八进制数 $a_n a_{n-1} \cdots a_0 b_{-1} b_{-2} \cdots b_{-m}$ 的位是：

$a_n \times 8^n + a_{n-1} \times 8^{n-1} + \cdots + a_0 8^0 + b_{-1} \times 8^{-1} + b_{-2} \times 8^{-2} + \cdots + b_{-m} \times 8^{-m}$

其中 a_i 和 b_i 为 0，1，…，7 八个数码中一个。书写时在尾部加注字母 O，由于字母 O 与数字 0 容易混淆，八进制数亦常用尾标 Q 标识。

十六进制数由 0，1，2，3，4，5，6，7，8，9，A，B，C，D，E，F 十六个数码构成，其中数码 A，B，C，D，E，F 对应于十进制数 10，11，12，13，14，15，基数为 16，第 i 位的权为 16^i，运算逢十六进一。

书写时在尾部加注字母 H 或下标 16，描述以字母开始的十六进制数的前面应加一数码 0，以区别于字符串。十进制数可以省略尾注 D。

十六进制数 $a_n a_{n-1}\cdots a_0 b_{-1} b_{-2}\cdots b_{-m}$ 的值是：

$a_n\times16^n+a_{n-1}\times16^{n-1}+\cdots+a_0 16^0+b_{-1}\times16^{-1}+b_{-2}\times16^{-2}+\cdots+b_{-m}\times16^{-m}$

其中 a_i 和 b_i 为 0，1，…，F 十六个数码中一个。

例如：$14AFH=1\times16^3+4\times16^2+10\times16^1+15=5295_{10}$

十、二、八和十六进制数之间的对应关系见表 1-1。

表 1-1　十、二、八、十六进制对应关系表

十进制	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
二进制	000	001	010	011	100	101	110	111								
八进制	0	1	2	3	4	5	6	7	8							
二进制	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
十六进制	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

3. 数制转换

（1）非十进制数转换成十进制数

非十进制数转换成十进制数的方法很简单，只需按权展开后相加即可。

【例 1-1】

$101101.01B=1\times2^5+1\times2^3+1\times2^2+1+1\times2^{-2}=45.25_{10}$

$345Q=3\times8^2+4\times8^1+5\times8^0=229_{10}$

$0F2DH=15\times16^2+2\times16^1+13\times16^0=3885_{10}$

（2）十进制数转换成非十进制数

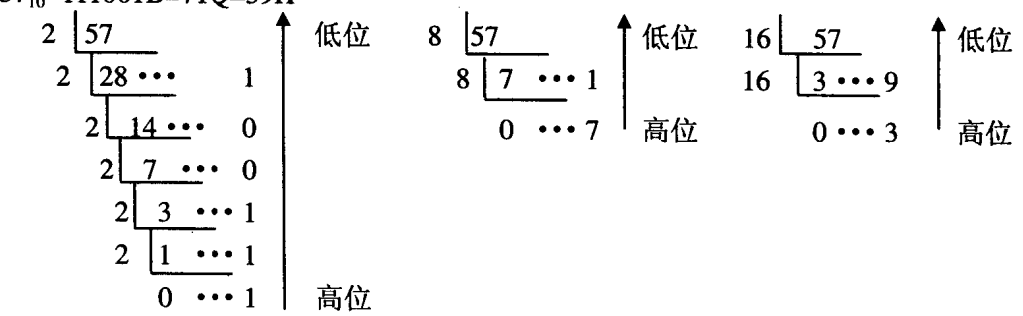
转换时将整数部分与小数部分分别转换。

整数部分采用除基数取余法，直至商为 0。先得到的余数为低位，后得到的余数为高位。

小数部分采用乘基数取整法，直至乘积为整数或达到控制精度。

【例 1-2】

$57_{10}=111001B=71Q=39H$



$0.625_{10}=0.101B=0.5Q=0.AH$

$0.625\times2=1.25\cdots1$　　$0.625\times8=5\cdots5$　　$0.625\times16=10\cdots A$

$0.25\times2=0.5\cdots0$

$0.5\times2=1\cdots\cdots1$

（3）二、八、十六进制数之间的转换

将二进制数转换成八进制数可按三位一组进行，转换成十六进制可按四位一组进行，每一组对应八进制或十六进制相应数码。分组时如果位数不够，整数部分在最左边补 0，小数部分在最右边补 0。

将八进制数转换成二进制数，只需将八进制数一位对应换成二进制数三位即可；同样对十六进制数，只需将其中一位对应换成二进制数四位即可。

【例 1-3】

1100100.11010B=001 100 100.110 100=144.64Q
　　1　4　4　6　4

=0110 0100.1101 0000=64.DH
　6　4　D　0

1.1.2 计算机中的数据表示

数的原值称为真值，它用常规的数符正负号及常规的数码表示。

在计算机中，数的符号也用二进制来表示。连数符一起数字化了的数，称为机器数。一般用最高有效位表示数的符号，正数用 0 表示，负数用 1 表示。机器数可用不同码制表示，常用的有原码、补码、反码三种表示法，广泛应用的是原码和补码两种表示法。

1. 数的原码表示

原码表示是一种比较直观的机器数表示方法，原码与真值的区别仅仅是符号位数字化。+1011 采用原码表示为 01011，-1011 采用原码表示 11011。数据 0 的原码有两种，即正 0 和负 0。

[+0]=000…000

[−0]=100…000

采用原码做乘除运算比较方便，可取绝对值直接运算，而符号单独处理。对于应用最多的加减运算，原码表示不太方便，像(−2)+3 表面做加法操作，而实际需做(3−2)减法操作；在计算机中实现时，比较繁琐。

2. 数的补码表示

为了简化运算，人们总结出数的另一种表示方法——补码表示法。

正数的补码就是其本身，形式与原码相同。负数的补码符号与原码相同，其余各位取反，然后末位加 1。

+1011 采用补码表示为 01011，-1011 采用补码表示为 10101。

$[x-y]_{补}=[x]_{补}+[-y]_{补}$，符号位同时参加运算，所以补码表示可以将减法变成加法。

从补码求原码或真值，可按求补码同样的方法进行。

n 位补码表示的数的范围为$[-2^{n-1}, 2^{n-1}-1]$。

【例 1-4】 求-59+61 的值。

$(-59)_{10}+(61)_{10}$=1100 0101+0011 1101=1 0000 0010=$(2)_{10}$

溢出

不仅二进制数有补码的概念，任意进制数都可以引入补码的概念。R 进制数的补码定义为：

$[x]_{补}=r^n+[x]$

【例 1-5】 求十进制数-41 和 59 的补码。

$[-41]_{补}=10^2+(-41)=59$

$[59]_{补}=10^2+59=159=59$

溢出

3. 数的反码表示

正数的原码与原码相同，负数的反码最高位为 1，其余各位依次求反。

由反码的定义可以得出补码的定义：正数的补码是它本身，负数的补码是它的反码加 1。

在反码表示中，“0”的表示法也不惟一。

$[+0]_{反}$=0000 0000　　$[-0]_{反}$=1111 1111

4. BCD 码

十进制数的二进制编码称为 BCD 码。它的引入是为解决日常习惯的十进制与机器内的二进制之间的矛盾，方便进行十进制数与二进制数之间的转换，最常用的是 8421 码。它对每一位十进制数码用 4 位二进制编码表示。表示规则见表 1-2。

表 1-2　BCD 码编码规则

十进制数	0	1	2	3	4	5	6	7	8	9
BCD 码	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

BCD 码在 0 至 9 的范围与纯二进制无区别，不同之处仅在于逢十进位，而 4 位纯二进制数逢十六进位，因此对大于等于 10 的数需做加 6 修正。

十进制数 1329 用 BCD 码表示为：0001 0011 0010 1001

BCD 码有压缩 BCD 码和非压缩 BCD 码两种形式。压缩 BCD 码的每个十进制数字按四个二进制位一组存放，一个字节可以存放两位压缩 BCD 码。非压缩 BCD 码的每个十进制数字占每个字节的低 4 位，高 4 位内容是什么不重要，每个字节存放一位 BCD 码。

【例 1-6】 写出十进制数 31 的压缩 BCD 码和非压缩 BCD 码。

31 的压缩 BCD 码为：0011 0001

31 的非压缩 BCD 码为：0000 0011 0000 0001

5. ASCII 码

ASCII 码是国际上比较通用的字符二进制编码，它的全称为美国信息交换标准码(American Standard Code for Information Interchange)。ASCII 码是 7 位二进制编码，表 1-3 列出了 ASCII 码。

从表 1-3 可看到，它对 94 个常用的一般符号进行了编码，其中包括 26 个英文字母的大小写符号、10 个数字符号和 32 个其他符号。空格也作为一个符号，其编码是 20H，它介于一般符号和 32 个控制符之间。所有这些一般符号和控制符统称为字符。从表 1.3 还可看到，数字符号的编码、大写字母符号的编码和小写字母符号的编码分别是连续的，所以只在记住数字符号的编码从 30H 开始、大写字母符号的编码从 41H 开始和小写字母的编码从 61H 开始，那么就可推出其他数字符号和字母符号的编码。

由于 ASCII 码只使用了 7 位二进制进行编码，故最多表示 128 个字符。这往往不能满足使用要求。为此，在 IBM PC 系列及其兼容机上使用扩展的 ASCII 码。扩展的 ASCII 码

使用 8 位二进制进行编码，故可表示 256 个字符。另外，在扩展的 ASCII 码中，控制符所对应的编码同时也表示其他图形符号。

表 1-3 ASCII 表

低位 \ 高位		000	001	010	011	100	101	110	111
		0	1	2	3	4	5	6	7
0000	0	NUL	DEL	SP	0	@	P	'	p
0001	1	SOH	DC1	!	1	A	Q	a	q
0010	2	STX	DC2	"	2	B	R	b	r
0011	3	ETX	DC3	#	3	C	S	c	s
0100	4	EOT	DC4	$	4	D	T	d	t
0101	5	ENQ	NAK	%	5	E	U	e	u
0110	6	ACK	SYN	&	6	F	V	f	v
0001	7	BEL	ETB	'	7	G	W	g	w
1000	8	BS	CAN	(	8	H	X	h	x
1001	9	HT	EM	)	9	I	Y	i	y
1010	A	LF	SUB	*	:	J	Z	j	z
1011	B	VT	ESC	+	;	K	[	k	{
1100	C	FF	FS	'	<	L	\	l	|
1101	D	CR	GS	-	=	M	]	m	}
1110	E	SO	RS	.	>	N	^	n	~
1111	F	SI	US	/	?	O	-	o	DEL

6. 变形国标码

有了 ASCII 码，计算机就能处理数字、英文字母等字符，但不能处理汉字。为使计算机能够处理汉字信息，就必须对汉字进行编码。我国在 1981 年 5 月对六千多个常用汉字制定了交换码的国家标准，即 GB2312-80《信息交换用汉字编码字符集——基本集》。该标准规定了汉字信息交换用的基本汉字符和一般图形字符，共计 7445 个。其中汉字分成两级，共计 6763 个。该标准同时也给定了它们的二进制编码，即国标码。

国标码是 16 位编码，高 8 位表示汉字符的区号，低 8 位表示汉字符的位号。实际上，为了给汉字符编码，该标准把代码表分成 94 个区，每个区有 94 个位。区号和位号都从 21H 开始。一级汉字安排在 30H 区至 57H 区，二级汉字安排在 58H 区至 77H 区。例如，“啊”字的国标码是 3021H。国标码为汉字的输入提供一种标准输入方式。

目前，计算机中文平台中普遍采用汉字编码是变形国标码。变形国标码是 16 位二进制编码，顾名思义它是国标码的变形。用得最多的变形方法是把国标码的第 15 位和第 7 位均置成 1，以区别 ASCII 码。由于国标码中第 15 位和第 7 位都是 0，所以这种变形方法实际上就是在国标码上加 8080H。

尽管 16 位的变形国标码与两个扩展的 ASCII 码的组合有冲突，但它在相关系统模块的支持下，有效地实现了汉字在计算机内的表示。

1.1.3 基本数据类型

计算机存取的以二进制位表示的信息位数一般是8的倍数，它们有专门的名称。

1. 字节

一个字节由8个二进制位组成。字节的最低位一般称为第0位，最高位称为第7位。

通常，硬件存储器的每个存储单元由8个连续的位组成，可用于存储一个字节的信息。

若用一个字节来表示一个无符号数，那么表示范围是0～255；如果表示有符号数，则表示范围是-128～+127。一个字节足以表示一个ASCII字符，也可以表示一个扩展的ASCII字符。

一个字节可分成2个4位的位组，称为半字节。高4位称高半字节，低4位称低半字节。

2. 字

两个字节（即16个二进制位）组成一个字。字的最低位称为第0位，最高位称为第15位。字的低8位称为低字节，高8位称为高字节。由于一个字由16个二进制位组成，所以用一个字来表示无符号数，则表示范围是0～65535；如果表示有符号数，则表示范围是-32768～+32767间的有符号数。一个字还可表示一个变形国标码。

注意，有时候字是涉及处理器一次能够处理的信息量的一个术语，字长是衡量处理器品质的一个重要指标。

3. 双字

就和听起来一样，双字由两个字组成，也即包含32个二进制位。低16位称为低字，高16位称为高字。双字表示的数的范围更大。

4. 四字

四字就是由四个字组成的，包含64个二进制位。如果双字不能表达所需的数值精度，那么四字也许就能解决问题了。

5. 十字节

十字节就和它的名称一样，由10个字节组成，含80个二进制位。可用于存储非常大的数或表示较多的信息。

6. 字符串

字符串是指由字符构成的一个线性数组。通常每个字符用一个字节表示，但有时每个字符也可用一个字或一个双字来表示。

1.2 汇编语言程序设计

1.2.1 程序设计语言

1. 指令与程序

指令是指出计算机所要进行的操作和操作对象的一组代码，实际上是计算机能够识别

的一组二进制代码。计算机中的指令由操作码和操作数组成，操作码说明计算机要进行的操作，操作数说明操作的对象。

操作码在机器里比较简单，只需对每一种操作指定确定的二进制代码就可以了。操作数字段比较复杂，首先它可以是一个、两个或三个，分别称为一地址、二地址或三地址指令；有的指令没有地址，称为无地址指令。其次操作数可能存放不同的地方，既可存放在寄存器中，也可能存放在存储器中。

一台计算机全部指令的集合，构成该计算机的指令系统。指令系统是计算机基本功能的体现。不同的计算机有不同的指令系统。

程序是使计算机完成工作，实现预期目的而用程序设计语言设计的一系列操作，体现形式是指令序列。

2. 程序设计语言

机器语言 由于计算机硬件本身只能识别 0、1 形式的二进制代码，因此计算机发展初期人们使用机器码（二进制码）来编写程序。这种二进制编码的计算机语言即机器语言。机器语言描述的程序称为目的程序或目标程序。只有目标程序 CPU 才能直接执行。机器语言没有明显的特征，难于记忆和理解，编程序时既麻烦又容易出错，也不便于学习。除用于编写机器的核心程序外，在实际中很少直接使用机器语言。

机器语言是面向机器的，一种机器有一种机器语言，不同的机器语言不通用。

汇编语言 汇编语言是一种采用助记符表示的程序设计语言。即用助记符表示指令的操作码和操作数，用标号或符号表示地址、常量和变量。助记符一般都是英语词的缩写，因而方便人们书写、阅读和检查。例如，用 ADD 表示加法，用 MUL 表示乘法等等。

用汇编语言书写的程序叫汇编语言源程序。汇编语言源程序必须翻译成机器语言程序才能被机器识别和运行，这个翻译过程叫汇编。汇编可以分为手工汇编和机器汇编，手工汇编指程序员手工完成汇编语言程序的翻译，机器汇编指利用计算机程序完成汇编语言源程序的翻译工作。具有将汇编语言源程序翻译成机器码功能的程序称为汇编程序。

汇编语言是面向机器的，即汇编语言的指令与机器语言指令是一一对应的。汇编语言依赖于具体机器，CPU 不同，相应的汇编语言就不同。汇编语言程序不能在不同的机器上通用。汇编语言和机器语言都是低级语言。

面向过程的高级语言 为了更加易学易用，提高程序的通用性，人们设计了大量接近自然语言的程序设计语言，这些语言面向用计算机求解问题的过程，不依赖具体机器，与特定机器相分离，采用接近自然语言的词汇。典型的高级语言有 BASIC、Pascal、C、FORTRAN、COBOL 语言等。

面向对象的高级语言 程序设计语言的更高形式是面向求解问题本身的高级语言。典型的面向对象的语言有 C++、Smalltalk、面向对象的 Pascal 语言等。

1.2.2 如何学习汇编语言

1. 汇编语言的特点

由于汇编语言使用指令助记符和符号地址，所以它要比机器语言容易掌握得多，与高级语言相比较，汇编语言有如下 4 个特点。

（1）汇编语言与机器关系密切