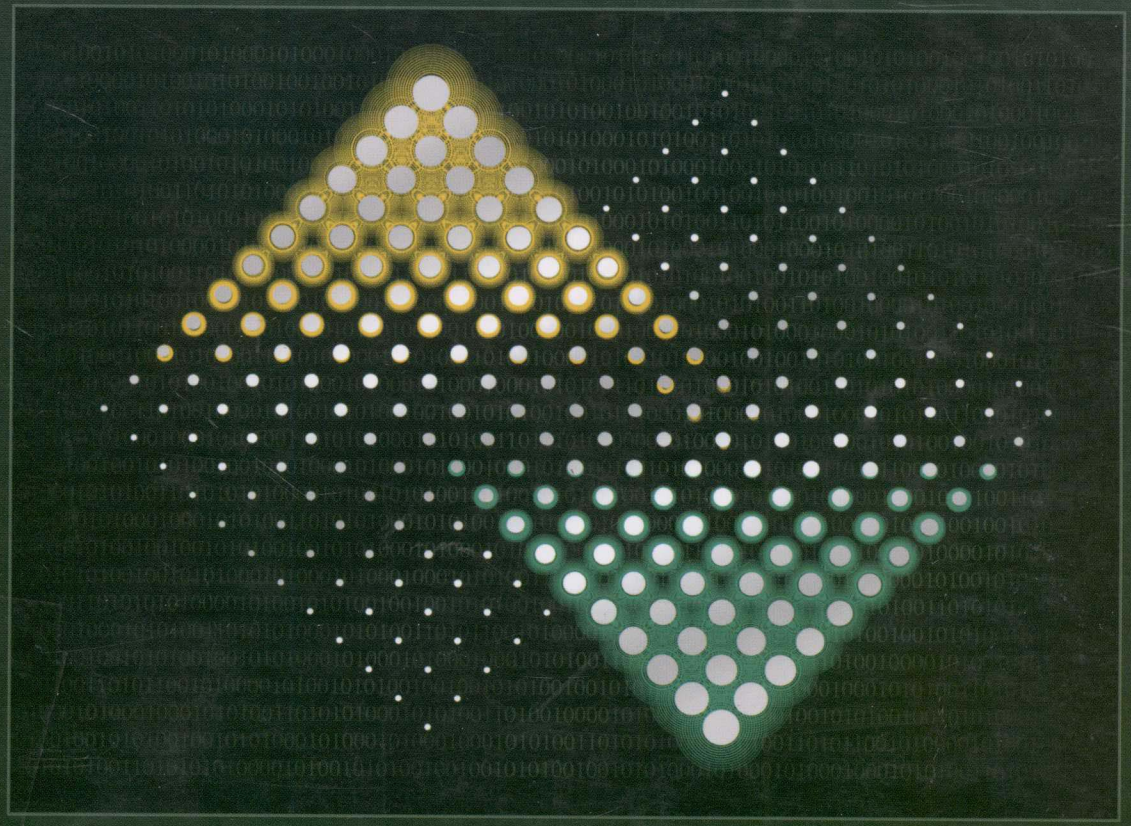
新编计算机类本科规划教材

汇编语言程序设计

（第3版）

徐建民　邵艳华　编著



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

新编计算机类本科规划教材

汇编语言程序设计

（第3版）

徐建民　邵艳华　编著

電子工業出版社
Publishing House of Electronics Industry
北京 • BEIJING

内 容 简 介

本书内容分为 12 章。第 1 章介绍汇编语言程序设计的基础知识，第 2 章介绍微处理器的结构及存储器的组成，第 3 章介绍寻址方式和指令系统，第 4 章介绍伪指令及汇编语言源程序结构，第 5 章介绍基本结构程序设计，第 6 章介绍子程序设计，第 7 章介绍实模式下的中断程序设计，第 8 章介绍输入/输出程序设计，第 9 章介绍高级汇编技术，第 10 章介绍保护模式概述，第 11 章介绍保护模式下的程序设计，第 12 章介绍保护模式下的中断和输入/输出。本书每章后都配有习题，并提供免费电子课件。

本书适合作为高等院校计算机及相关专业汇编语言程序设计课程的教材或教学辅导书，也可作为希望掌握 Windows 汇编程序设计的中高级程序开发人员的自学参考书。

图书在版编目（CIP）数据

汇编语言程序设计 / 徐建民，邵艳华编著. —3 版. —北京：电子工业出版社，2010.6
新编计算机类本科规划教材
ISBN 978-7-121-08045-6

I. 汇… II. ①徐… ②邵… III. 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（2008）第 210378 号

责任编辑：冉 哲
印 刷：北京东光印刷厂
装 订：三河市皇庄路通装订厂
出版发行：电子工业出版社
北京市海淀区万寿路 173 信箱 邮编 100036
开 本：787×1 092 1/16 印张：18.75 字数：480 千字
印 次：2010 年 6 月第 1 次印刷
印 数：3000 册 定价：29.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前　言

汇编语言程序设计一直是计算机及相关专业一门重要的核心课程。

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，因此作为计算机及相关专业的学生，有必要掌握汇编语言程序设计的基本方法。

汇编语言程序设计的相关知识是某些后续课程的基础。在某些后续课程（如操作系统、计算机接口技术等）的学习中，或多或少需要用到汇编语言程序设计的知识。

学习汇编语言程序设计课程，有利于学生更深入、全面地掌握计算机的工作原理，例如，可以更好地理解高级程序设计语言中的分支、循环结构，还可以更好地理解存储器、地址等有关知识等。

2001 年，根据全国高等学校计算机教育研究会课程与教材建设编委会的意见，作者编写了本教材；2005 年，在原教材的基础上，作者又进行了修订，编写了第 2 版教材。与前两版教材相比，第 3 版教材主要做了以下修订：

1．为大多数例题增加了流程图及分析，更方便读者学习。

2．对个别章节内容进行了调整，更加注重基本方法的讲解。

3．重新编写了保护模式程序设计部分，使其内容更容易理解。

本书内容分为 12 章。第 1 章介绍汇编语言程序设计的基础知识；第 2 章介绍微处理器的结构及存储器的组成；第 3 章介绍寻址方式和指令系统；第 4 章介绍伪指令及汇编语言源程序结构；第 5 章介绍基本结构程序设计；第 6 章介绍子程序设计；第 7 章介绍实模式下的中断程序设计；第 8 章介绍输入/输出程序设计；第 9 章介绍高级汇编技术；第 10 章介绍保护模式概述；第 11 章介绍保护模式下的程序设计；第 12 章介绍保护模式下的中断和输入/输出。本书每章后都配有习题，并提供免费电子课件，请登录华信教育资源网下载，网址：http://www.hxedu.com.cn。

本书适合作为高等院校计算机及相关专业汇编语言程序设计课程的教材或教学辅导书，也可作为希望掌握 Windows 汇编程序设计的中高级程序开发人员的自学参考书。

全书由河北大学徐建民负责整体策划，并执笔修订了其中的第 1～9 章，贵州民族大学的邵艳华编写了第 10～12 章，修订了全书的程序流程图。全书完成后由邵艳华做了第一遍修改，再由徐建民进行了统稿。由于作者的精力和水平有限，本教材肯定还有许多不足之处，恳请读者不吝赐教，使本教材得到改正与完善。

本书的修订得到了许多同仁和朋友的热情帮助，恕不一一列举，在此一并表示感谢。

作　者

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，习惯上常采用十进制计数法。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：

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

式中，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。为了与十进制数相区别，书写时，在其尾部加注字母 B 或下标 2。

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

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

式中，a_i 和 b_i 为 0 或 1。

例如：$101011B = 1\times2^5 + 0\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 = 43$

n 位二进制数可以表示 2^n 种组合，3 位能表示 8 种组合，即 0～7 的整数，4 位能表示 16 种组合，即 0～15 的整数，8 位能表示 256 种组合，即 0～255 的整数，16 位能表示 64×1024（64KB）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，阅读、书写、记忆等不太方便。例如十进制数 1000，用二进制数表示则需要 10 位二进制数字 1111101000B。为了便于阅读与书写，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 共 8 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 $a_n a_{n-1} \cdots a_0 \cdot b_{-1} b_{-2} \cdots b_{-m}$ 的值是：

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

式中，a_i 和 b_i 为 0，1，…，7 这 8 个数码中的任一个。

书写八进制数时，在其尾部加注字母 O 或下标 8。由于字母 O 与数字 0 容易混淆，因此，八进制数常用尾注 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 共 16 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。

书写十六进制数时，在其尾部加注字母 H 或下标 16，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 或下标 10。

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

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

式中，a_i 和 b_i 为 0，1，…，F 这 16 个数码中一个。

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

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、十六进制数对应关系表

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

3．数制转换

（1）非十进制数转换为十进制数

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】非十进制数转换为十进制数举例。

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

$345Q=3\times8^2+4\times8^1+5\times8^0=229$

$0F2DH=15\times16^2+2\times16^1+13\times16^0=3885$

（2）十进制数转换为非十进制数

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 为止。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为止。

【例 1-2】十进制数转换为非十进制数举例。

57=111001B=71Q=39H

2 | 57	低位
2 | 28 … 1	
2 | 14 … 0	
2 | 7 … 0	
2 | 3 … 1	
2 | 1 … 1	
0 … 1	高位

8 | 57	低位
8 | 7 …1	
0 …7	高位

16 | 57	低位
16 | 3 …9	
0 …3	高位

0.625=0.101B=0.5Q=0.AH

0.625×2=1.25…1　　0.625×8=5…5　　0.625×16=10…A

0.25×2=0.5……0

0.5×2=1……1

（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 3 位一组进行，转换成十六进制数可按 4 位一组进行，每一组对应八进制数或十六进制数相应的数码。分组时，若位数不够，则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，只需将八进制数的 1 位对应转换成二进制数 3 位即可。同样，对十六进制数，只需将其中的 1 位对应转换成二进制数的 4 位即可。

【例 1-3】转换举例。

1100100.11010B=001 100 100.110 100=144.64Q
　　1　4　4　6　4
=0110 0100.1101 0000=64DH
　　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数来表示。连同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。机器数可以用不同码制表示，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 采用原码表示为 11011。数据 0 的原码有两种，即正 0 和负 0。

[+0]=000…000

[−0]=100…00

采用原码进行乘除运算比较方便，可取绝对值直接运算，而符号单独处理。但对于应用最多的加减运算，原码表示不太方便，例如，（−2）+3 从表面上看是加法操作，而实际需进行 3−2 减法操作，在计算机中实现时，比较烦琐。

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。

补码减法运算为：

$$[x-y]_{补}=[x]_{补}+[-y]_{补}$$

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】求十进制数−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。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

正数的反码与原码相同，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，数据 0 的表示法也不唯一。

$[+0]_{反}$=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数与机器内的二进制数之间的矛盾，并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。它把每 1 位十进制数码用 4 位二进制编码表示。编码规则见表 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 0～9 的范围内与纯二进制数无区别，不同之处仅在于，BCD 码逢十进位，而 4 位纯二进制数逢十六进位，因此对大于等于 10 的数需作加 6 修正。

例如，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 4 个二进制位一组存放，一个字节可以存放 2 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 1 位 BCD 码。

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

十进制数 31 的压缩 BCD 码为：0011 0001

十进制数 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 中列出了 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，所以只要记住

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，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 位二进制数进行编码，故最多表示 128 个字符。但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 位二进制数进行编码，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码表

高位 / 低位		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，但不能处理汉字字符。为了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 6000 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～57H 区，二级汉字安排在 58H～77H 区。例如，“啊”字的国标码是 3021H。国标码为汉字的输入提供了一种标准输入方式。

目前，计算机中文平台中普遍采用的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 以区别于 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，硬件存储器的每一存储单元都由 8 个连续的位组成，可用于存储一个字节的信息。

例如，用一个字节来表示一个无符号数，那么表示范围是 0～255；而表示有符号数，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。

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

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；而表示有符号数，则表示范围是–32768～+32767。一个字还可表示一个变形国标码。

注意，有时，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

顾名思义，双字由 2 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。双字能表示的数的范围更大。

4. 四字

四字就是由 4 个字组成，包含 64 个二进制位。如果双字还不能表达所需要的数值精度，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 汇编语言程序设计概述

1.2.1 程序设计语言

1. 指令与程序

指令是用于指出计算机要进行的操作和操作对象的一组代码，实际上是计算机能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。

操作码字段在机器里比较简单，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，首先它可以有一个、两个或三个操作数，分别称为一操作数指令、二操作数指令或三操作数指令，还有的指令没有操作数，称为无操作数指令；其次，操作数可能存放在不同的地方，既可以存放在寄存器中，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。

程序是为使计算机完成工作，实现预期目的，而用程序设计语言编写的一系列操作，其体现形式是指令序列。

2. 程序设计语言

（1）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，因此在计算机发展初期，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语言就是机器语言。机器语言描述的程序称为目的程序或目标程序，只有目标程序才能由 CPU 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，不同的机器之间语言不通用。

（2）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，即用助记符来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单词的缩写，因而便于人们书写、阅读和检查。例如，用 ADD 表示加法，用 MUL 表示乘法等。

用汇编语言书写的程序叫做汇编语言源程序。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，这个翻译过程叫做汇编。汇编可以分为手工汇编和机器汇编两类。手工汇编是指程序员手工完成汇编语言程序的翻译，机器汇编是指利用计算机程序完成汇编语言源程序的翻译工作。用于将汇编语言源程序翻译成机器语言的程序称为汇编程序。

汇编语言的指令与机器语言指令是一一对应的，因此，汇编语言是面向机器的。CPU 不同，相应的汇编语言就不同。汇编语言和机器语言都是低级语言。

（3）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 和 COBOL 等。

（4）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 和面向对象的 Pascal 等。

1.2.2 汇编语言的特点和使用场合

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

1. 汇编语言与机器关系密切

汇编格式指令是机器指令的符号表示，汇编格式指令与机器指令是一一对应的，因此它与机器有着密切的关系。不同类型的 CPU，有不同的汇编语言，也就有各种不同的汇编程序。

由于汇编语言与机器关系十分密切，所以汇编语言源程序与高级语言源程序相比，它的通用性和可移植性要差得多。但是，通过汇编语言可以最直接和最有效地控制机器，这常常是大多数高级语言难以做到的。

2. 汇编语言程序效率高

用汇编语言编写的源程序在汇编后所得到的目标程序效率高。这种目标程序的高效率反映在时间和空间两个方面，即运行速度快、目标程序短。在采用相同算法的前提下，任何高级语言程序在这两方面的效率都不如汇编语言程序，在许多情况下更是远远不及。

汇编语言能获得时间和空间高效率的主要原因是：构成汇编语言主体的汇编格式指令是用机器指令的符号表示的，每一条汇编格式指令都对应某条机器指令；汇编语言程序能直接并充分利用机器硬件系统的许多特性。高级语言程序在上述两点上要逊色得多。

3. 编写汇编语言源程序烦琐

编写汇编语言源程序要比编写高级语言源程序烦琐得多。

机器指令符号化的每一条汇编格式指令所能完成的操作极为有限。例如，Z80 指令集中没有乘法指令，8086 指令集中没有能够同时完成两次算术运算的指令。

在用汇编语言编写程序时，必须考虑包括寄存器、存储单元和寻址方式在内的几乎所有的细节问题，例如，指令执行对标志位的影响，堆栈设置的位置等。在使用高级语言编写程序时，这些问题由操作系统完成，程序员不会遇到这些琐碎却很重要的问题。

4. 汇编语言程序调试困难

调试汇编语言程序往往要比调试高级语言程序困难。汇编格式指令的功能有限以及程序员要注意太多的细节问题是造成这种困难的两个客观原因；汇编语言为程序员提供了最大的舞台，而程序员往往为了追求时间和空间上的高效率而不顾程序的结构，这是造成调试困难的主观原因。

由于汇编语言的特点，下列应用场合可考虑使用汇编语言。

① 对软件的执行时间或存储容量有较高要求的场合，例如，系统程序的关键核心，智能化仪器仪表的控制系统和实时控制系统等。

② 需要提高大型软件性能的场合。通常，大型软件中执行频率高的子程序（过程）用汇