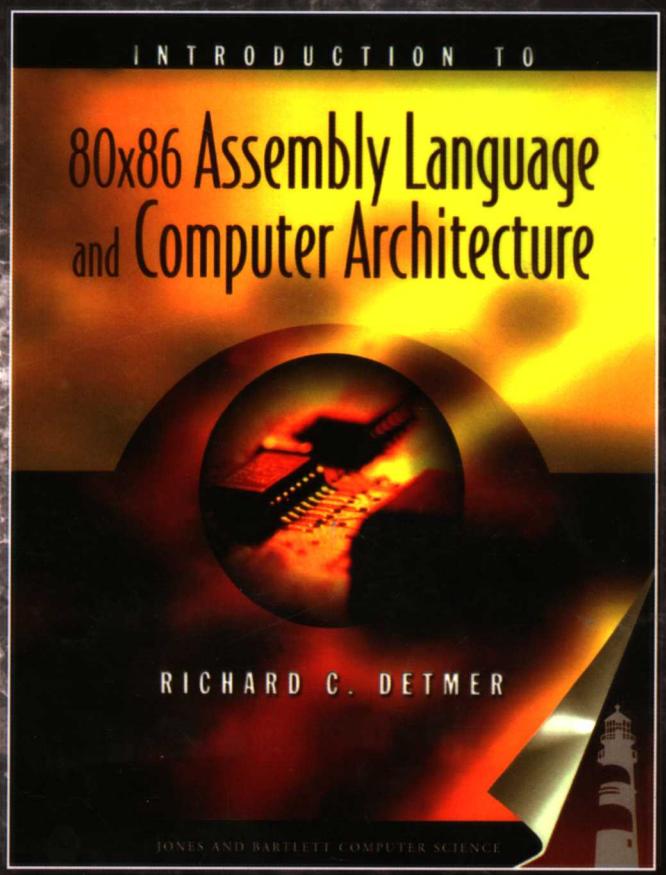
Jones and Bartlett

计 算 机 科 学 丛 书

80x86汇编语言与计算机体系结构

(美) Richard C. Detmer 著　郑 红　庞毅林　蒋翠玲　译



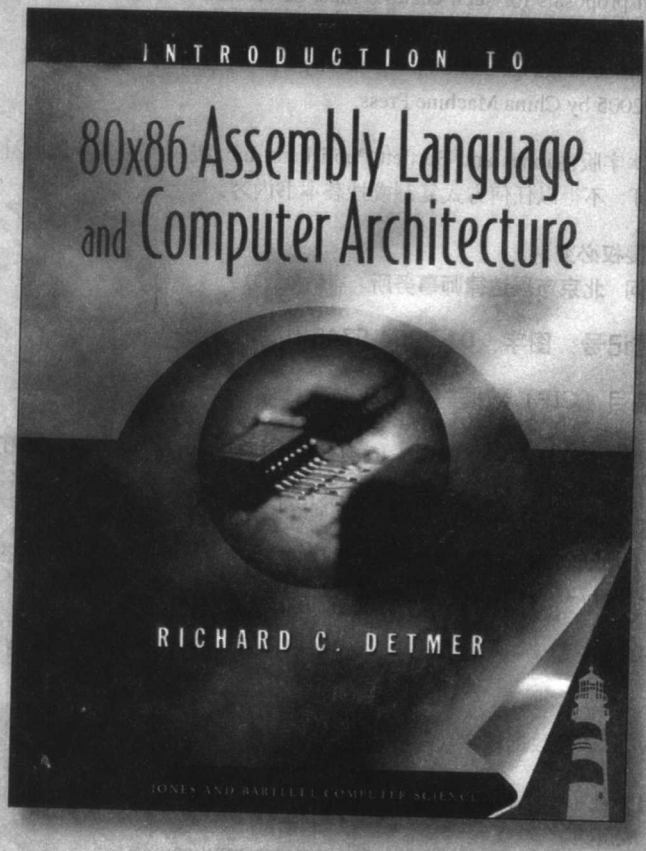
Introduction to 80x86 Assembly Language and Computer Architecture

机械工业出版社
China Machine Press

计 算 机 科 学 丛 书

80x86汇编语言与计算机体系结构

(美) Richard C. Detmer 著　郑红　庞毅林　蒋翠玲　译



Introduction to 80x86 Assembly Language and Computer Architecture

机械工业出版社
China Machine Press

本书在当前操作系统采用的平面32位地址环境中介绍了80x86汇编语言和计算机体系结构，重点介绍32位平面内存模型，强调了体系结构的概念，如寄存器、内存编址、硬件功能等，涵盖了汇编语言的指令、分支和循环、过程、位运算、汇编过程、输入/输出等重点内容，并增加了高级语言的概念，同时理论结合实例，注重关键知识点练习与编程实践。

本书适合作为高等院校相关专业的教材以及参考书，也可供工程技术人员参考。

Richard C. Detmer: Introduction to 80x86 Assembly Language and Computer Architecture (ISBN 0-7637-1773-8).

本书版权登记号：图字：01-2004-5745

图书在版编目（CIP）数据

80x86汇编语言与计算机体系结构 /（美）戴默（Detmer, R. C.）著；郑红等译. -北京：机械工业出版社，2006.1

（计算机科学丛书）

书名原文：Introduction to 80x86 Assembly Language and Computer Architecture

ISBN 7-111-17617-0

Ⅰ.8… Ⅱ.① 戴… ② 郑… Ⅲ.① 汇编语言 ② 计算机体系结构 Ⅳ.①TP313 ② TP303

中国版本图书馆CIP数据核字（2005）第125410号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：范运年

北京瑞德印刷有限公司印刷 · 新华书店北京发行所发行

2006年1月第1版第1次印刷

787mm × 1092mm 1/16 · 21.5印张

印数：0 001 - 4 000册

定价：49.00元（附光盘）

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

本社购书热线：（010）68326294

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭橥了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall，Addison-Wesley，McGraw-Hill，Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum，Stroustrup，Kernighan，Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及庋藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师们服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业的教学度身订造的。其中许多教材均已为M. I. T.，Stanford，U.C. Berkeley，C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

电子邮件：hzjsj@hzbook.com
联系电话：（010）68995264
联系地址：北京市西城区百万庄南街1号
邮政编码：100037

专家指导委员会

（按姓氏笔画顺序）

尤晋元	王　珊	冯博琴	史忠植	史美林
石教英	吕　建	孙玉芳	吴世忠	吴时霖
张立昂	李伟琴	李师贤	李建中	杨冬青
邵维忠	陆丽娜	陆鑫达	陈向群	周伯生
周克定	周傲英	孟小峰	岳丽华	范　明
郑国梁	施伯乐	钟玉琢	唐世渭	袁崇义
高传善	梅　宏	程　旭	程时端	谢希仁
裘宗燕	戴　葵			

译 者 序

关于这本书的意义和它的主要内容，本书的作者在前言中已经讲得很详细了。我只想简单谈谈在翻译过程中的一些感想。

首先，我很高兴有机会翻译这本书，因为，在过去我们学习计算机课程时，汇编语言课程是使用单独的教材，很少和计算机体系结构结合在一起，学习汇编语言常常令人觉得有些枯燥，并且似乎有些难学易忘；同样，在教授计算机体系结构时，也没有过多地讨论汇编语言程序设计。但这本书很好地将软件设计与硬件结构知识融合在一起，通过一些精选的实例，由浅入深地介绍了汇编语言程序设计的特点以及计算机的工作。因此，通过翻译这本书，不仅让我重温了这两门课程，而且更深层次地理解了计算机的体系结构。

其次，我要感谢机械工业出版社对我的信任，在对书稿的处理过程中，诸位编辑给予了很多帮助，特别是范运年编辑和朱起飞编辑反复征询译者的意见，对本书的译稿提出了许多宝贵的建议。此外，文欣秀老师和朱法枝老师对本书翻译中遇到的个别问题，提出了中肯的意见，在此一并表示感谢。

最后，我要感谢我的家人，他们的支持和鼓励使我能够完成翻译工作。尤其是我的孩子，刚开始翻译时，他尚未出生，他还在孕育中就陪我一起经过了初稿阶段。此后，尽管我常常因为校稿要把他放在一边，减少了对他的照顾，但是，只要我离开电脑向他走去，他总是用最开心、最灿烂的笑容迎接我。

本书的第1章、第4章、第7章由庞毅林翻译，蒋翠玲参与了第9章的翻译，其余章节主要由郑红翻译，全书最后由郑红和庞毅林统稿。由于译者水平所限，加之时间仓促，译文中难免有不妥之处，恳请广大读者不吝批评指正。

译 者

2005年11月

前　言

计算机可以从多种不同的层次来认识。有些人只对字处理或者游戏之类的计算机应用软件感兴趣，但是，计算机程序员通常把计算机作为一个工具，用来编写新的应用软件。通过语言编译器，高级语言程序员更深入地认识了计算机，编译器给人的印象是，计算机的内存地址中存储integer、real和array of char等等对象类型，计算表达式的值，调用过程，执行while循环等等。

然而，事实上计算机是在很低的层次上工作。本书强调计算机的体系结构层，也就是，由机器指令所定义的层次，处理器可以在该层执行。汇编语言指令直接翻译为机器语言指令，这样，当编写一个汇编语言程序时，就可以理解计算机在机器语言级是如何工作的。

尽管本书强调的是计算机操作的汇编语言/机器语言层，但也可从其他层次来认识计算机。本书讨论了高级语言中的一些概念，例如if语句在机器层是如何实现的。本书还讨论了操作系统的一些功能，并简要描述了在硬件层用到的逻辑门。另外，本书考察了汇编语言是如何翻译为机器语言的。

为了在任何层次都可以有效地编程，程序员必须了解在机器层的某些基本原理，它们在大多数的计算机体系结构中都要用到。本书将涉及以下基本概念:

- 存储地址，CPU寄存器及其使用
- 计算机中数值型格式的数据和字符串的表示
- 二进制补码整数的操作指令
- 单个位操作的指令
- 处理字符串的指令
- 分支和循环指令
- 过程编码：控制转移、参数传递、局部变量和调用程序的环境保护

本书中讨论的主要的计算机体系结构是大多数个人计算机所使用的80x86 CPU系列。但是，几乎每章都有其他体系结构，或者不同的计算机层次的信息。用汇编语言编程以及学习本书中的相关概念，有助于用任何编程语言进行有效的编程，激发对计算机设计和体系结构更进一步的研究，或者更多地了解某个特定的计算机系统的详细内容。

本书的组织结构和内容

本书中的大多数素材基于我的前一本书——《Fundamentals of Assembly Language Programming Using the IBM PC and Compatibles》。通过多年对这些素材的教学使我得出这样一个结论：对大多数学生而言，汇编语言课程是介绍计算机体系结构最好的课程。相对于编程而言，本书更多地强调体系结构。本书还重点介绍一些通用的概念，而不是某个特定的计算机系统的细节。

学习这门汇编语言课程要求的前提条件是至少要对高级语言结构有很好的理解。第3章～第6章及第8章是我第一学期课程的核心内容，第1章～第8章的内容我通常讲解得很详细，第9

章速度会快些，根据时间和可利用的资源，选择性讲解第10章～第12章的某些主题。例如，有时，我会通过某个C++程序中的汇编语句行来介绍浮点运算。

风格和教学

本书主要是例证教学。早在第3章本书就给出了一个完整的汇编语言程序，并且在学生能够理解的层次上，仔细地考察了程序的各个部分。随后的章节包含了许多汇编语言代码的例子，同时，对一些新的或者难以理解的概念给出了恰当的解释。

本书使用了大量的图表和例子。给出许多“指令执行前”和“指令执行后”的例子来讲解指令。本书还有一些演示调试程序（debugger）使用的例子。这些例子可以帮助学生深入了解计算机内部的工作。

每章的后面都有练习。答案简短的练习可以加深学生对学过的内容的理解，而且每章后面的编程练习也为学生提供了一个将书中的内容运用到汇编语言编程中的机会。

软件环境

“标准”的80x86汇编器是微软宏汇编器（MASM），版本为6.11。尽管该汇编器生成的代码用于32位的平面内存模式编程，非常适合Windows 95、Windows NT或者32位的微软操作系统环境，但是，与该软件包对应的链接器和调试程序并不适合在这样的系统环境中使用。本书附带一张光盘，包含MASM（ML）的汇编程序、最新的微软链接器、32位的全屏调试程序WinDbg（也来自于微软）以及必要的支持文件。该软件包为生成和调试控制台的应用程序提供了一个良好的环境。

本书配套光盘中不仅有本书的内容，也有可供学生使用的简单的输入/输出设计的软件包。因此，它强调的重点仍然是计算机体系结构而不是操作系统的细节。这个I/O包在本书中广泛使用。最后，该光盘还包含了每个程序的源代码，这些程序都会在书中出现。

致谢

我想感谢我的学生们，他们对本书的最初版本付出了很多努力，让我经常能及时地得到素材。这些学生非常善于捕捉错误。我也要感谢Hong Shi Yuan，在他的汇编语言课程上，他用了本书的最初版本，并提供了有价值的反馈意见。

我还要感谢花了很多时间来检查本书手稿的人们：Houston-Clear Lake大学的Dennis Bouvier、美国空军学院的Barry Fagin、Worcester工艺学院的Glynis Hamel、犹他谷州立大学的Dennis Fairclough、东南路易斯安娜大学的Thomas Higginbotham、Worcester工艺学院的Clifford Nadler。

我的妻子Carol值得称赞。当我在计算机前处理书稿时，经常忽略了她，而她都给予了理解。

Richard C. Detmer

目　录

第1章　计算机中数的表示

用Java或C++等高级语言编程时，要用到许多不同类型的变量（比如整型、浮点型或者字符型），变量一旦声明，就不需要考虑数据在计算机中是如何表示的。然而，用机器语言编程时，就必须考虑如何存储数据。因此，经常需要将数据从一种表示法转换为另一种表示法。本章将论述微型计算机中数的表示的几种常用方法。第2章概述微机的软件和硬件，第3章介绍如何编写汇编语言程序，由它直接控制计算机机器指令的执行。

1.1　二进制和十六进制数

计算机用位（bit，二进制数制中用不同的电子状态表示0或者1）来表示值。以2为基数，用数字0和1表示二进制数。二进制（binary）数跟十进制数很相似，只不过二进制相应的权（从右到左）依次为1、2、4、8、16（2的更高次幂等），而十进制相应的权为1、10、100、1000、10000（10的幂）。例如，二进制数1101可表示十进制数13。

1		1		0		1		
8	+	4	+	2	+	1	=	13

二进制数很长，因而在读写时很不方便，比如：八位二进制数11111010表示十进制数250，十五位二进制数111010100110000表示十进制数30000。而用十六进制（hexadecimal）（基数16）表示时，只需要用到相应二进制数表示的四分之一长度的位数。十六进制与二进制的转换很容易，因此，十六进制的表示可以缩短二进制的表示。十六进制需要十六个数字：其中0、1、2、3、4、5、6、7、8和9与十进制数相同；A、B、C、D、E、F等同于十进制的10、11、12、13、14和15。另外，这几个字母不论是小写还是大写都可用于表示数。

十六进制数中的权值对应16的幂，权值从右到左依次是1、16、256等等。十六进制数9D7A可如下计算得出表示的是十进制的40314：

```
  9      ×    4096    36864    [4096 = 16³]
+13      ×     256     3328    [D是13, 256 = 16²]
+ 7      ×      16      112
+10      ×       1       10    [A是10]
                =40314
```

表1-1给出了二进制、十六进制和十进制的关系。记住这张表，或者能够很快地建立这张表是很有必要的。

上面的两个例子显示了二进制数和十六进制数是如何转换为十进制数。那么如何将十进制数转换成二进制数或者十六进制数呢？以及如何将二进制数与十六进制数互相转换呢？随后的内容将介绍如何手工实现不同进制的转换。通常情况下，用一个具有二进制、十进制和十六进制转换功能的计算器很容易实现转换，这样，数制转换只不过是按一两个键而已。这

种计算器可像十进制那样进行二进制和十六进制的数学运算，而且具有很多其他的用途。注意：这种计算器很多都用七个显示段来显示一个数字。比如，显示小写字母b时看起来像数字6，其他的某些字符也有可能很难辨认。

表1-1 十进制、二进制和十六进制数的关系

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

不需要用计算器把十六进制数转换为对应的二进制形式。事实上，许多二进制数太长，一般的计算器不易显示。要进行转换，只要将每一个十六进制数用四位二进制数表示即可。其对应关系如表1-1的第3列所示。如果位数不够四位，必要的时候前面用0补充。例如：

$3B8E2_{16} = 11\ 1011\ 1000\ 1110\ 0010_2$

转换数字下标处的16和2表示基数。如果不会造成混淆，这些下标处的数字常可以忽略。二进制数补齐位数是为了增强可读性，如十六进制数2转换为二进制时，最前面用起始位0补齐得到0010。但由于二进制数前面的零不改变该二进制数的值，所以上例中十六进制数的最高位3不需要转换为0011。

把二进制数转换为十六进制数格式，则与上面的步骤正好相反。把二进制数从右向左每四位进行分隔，每四位二进制数用对应的十六进制数表示，例如：

$1011011101001101111_2 = 101\ 1011\ 1010\ 0110\ 1111_2 = 5BA6F_{16}$

前面介绍了如何将二进制数转换为十进制数，但一般不会将很长的二进制数直接转换为十进制数，更快的方法是先将二进制数转换为十六进制数，再将该十六进制数转换为十进制数。以上面的19位二进制数为例：

$$
\begin{aligned}
&1011011101001101111_2\\
&= 101\ 1011\ 1010\ 0110\ 1111_2\\
&= 5BA6F_{16}\\
&= 5\times 65536 + 11\times 4096 + 10\times 256 + 6\times 16 + 15\times 1\\
&= 375407_{10}
\end{aligned}
$$

下面将给出十进制数转换为十六进制数的算法，该算法从右到左依次生成十六进制数位。该算法用伪代码来描述，本书中其他的所有算法和程序都将采用伪代码描述。

```
until DecimalNumber = 0 loop
    divide DecimalNumber by 16, getting Quotient and Remainder;
    Remainder (in hex) is the next digit (right to left);
    DecimalNumber := Quotient;
end until;
```

例子:

以十进制数5876转换为十六进制数的过程为例:

- 因为这是一个until循环，当第一次执行程序体的时候就进行循环控制条件检查。
- 16整除5876（十进制数）

```
   367   商     新的十进制数的值
16)5876
   5872
   ----
      4  余数   最右边的十六进制数位
         当前结果: 4
```

- 367不等于0，再用16整除

```
    22   商     新的十进制数的值
16)367
   352
   ---
    15  余数   生成的第2个十六进制数位
        当前结果: F4
```

- 22不等于0，用16整除

```
     1   商     新的十进制数的值
16)22
   16
   --
    6  余数   生成的下一个十六进制数位
       当前结果: 6F4
```

- 1不等于0，用16整除

```
     0   商     新的十进制数的值
16)1
   0
   -
   1  余数   生成的下一个十六进制数
      当前结果: 16F4
```

- 当前的十进制数为0，循环终止。最后结果为$16F4_{16}$

在计算机中也用到八进制数（octal，基数为8）。八进制数用数字0～7表示，大多数计算器可进行十六进制和八进制运算。把二进制数的每3位转换为一个对应的八进制数，很容易实现二进制数八进制数的转换。同样，将八进制数转换为二进制数时，每一个八进制数用相应的3个二进制数位表示。要实现十进制转换为八进制，可以使用前面的十进制转换为十六进制的算法，只不过在做每一步时，用8而不是用16整除。

练习1.1

请根据给出的每一个数字将表中空白的另外两种进制形式补充完整。

	二进制	十六进制	十进制
1.	100	_____	_____
2.	10101101	_____	_____
3.	1101110101	_____	_____
4.	11111011110	_____	_____
5.	10000000001	_____	_____
6.	_____	8EF	_____
7.	_____	10	_____
8.	_____	A52E	_____
9	_____	70C	_____
10.	_____	6BD3	_____
11.	_____	_____	100
12.	_____	_____	527
13.	_____	_____	4128
14.	_____	_____	11947
15.	_____	_____	59020

1.2 字符编码

字母、数字、标点符号等各种字符在计算机中都是用特定的数值来表示的。字符编码方式有很多，微机中普遍采用的一种字符编码是美国信息交换标准代码（简称为ASCII，其发音为ASK-ee）。

ASCII用七位表示字符，数值从0000000～1111111，包括128个值，可以表示128种字符。也可用十六进制数00～7F或者十进制数0～127[⊖]表示。附录A给出了ASCII的详细列表，在表中可以查到“Computers are fun .”用十六进制表示的ASCII码值：

43	6F	6D	70	75	74	65	72	73	20	61	72	65	20	66	75	6E	2E
C	o	m	p	u	t	e	r	s		a	r	e		f	u	n	.

注意：尽管空格字符不可见，但仍然有一个字符编码（十六进制数20）

数字也可以用字符编码表示，例如用ASCII表示日期“October 21, 1976”：

4F	63	74	6F	62	65	72	20	32	31	2C	20	31	39	37	36
O	c	t	o	b	e	r		2	1	,		1	9	7	6

其中，日期中的数字字符21用ASCII码值32 31表示，1976用31 39 37 36表示，这与上节所讲的二进制表示有所不同，上节中$21_{10}=10101_2$，$1976_{10}=11110111000_2$。这两种方法在计算机中都可以表示数字：其中ASCII表示法用于外设输入输出，二进制表示法用于计算机内部计算。

ASCII码看起来似乎是任意指定的，但事实上是遵循某些规范的。大写字母的ASCII码是

⊖ 包括IBM及兼容系统在内的一些计算机，使用扩展的字符集，字符集增加了从十六进制数80～FF（十进制数128～255）的字符，本书中不使用扩展的字符集。

相邻的，同小写字母的ASCII码一样。大写字母的编码与其相对应的小写字母的编码仅仅有一位不同，大写字母的第5位是1，而小写字母的第5位是0，其他各位都相同。（通常计算机用位来表示数时，从右到左，最右边的位从第0位开始。）例如，

- 大写字母M的编码为$4D_{16} = 1001101_2$
- 小写字母m的编码为$6D_{16} = 1101101_2$

打印输出字符（printable character）从20_{16} ~ $7E_{16}$。（空格字符也是打印输出字符。）数字0、1、…、9的ASCII值分别为30_{16}、31_{16}、…、39_{16}。

ASCII码值从00_{16} ~ $1F_{16}$以及$7F_{16}$都是控制字符（control character），例如，ASCII键盘上的ESC键的ASCII码值是$1B_{16}$，简称ESC，表示特殊服务控制，但经常被认为是“escape”的含义。ESC字符经常与其他字符一起传给外部设备，比如，传给一台打印机，让它执行某种指定的操作。因为这样的字符序列没有标准化，所以本书将不作讨论。

本书中使用频率最高的两个ASCII控制字符是$0D_{16}$和$0A_{16}$，分别表示回车（CR）和换行（LF）。当按下ASCII键盘的Return或Enter键时，就会产生编码$0D_{16}$，如果该编码送到ASCII显示器，则使光标移到当前行的开始处而不是到新的一行；如果该编码送到ASCII打印机（至少是早期的一种打印机），则会使打印头移到当前行的开始处。换行码$0A_{16}$在ASCII显示器上会使光标垂直移到下一行或者使打印机将纸向上滚动一行。要想信息从新的一行的开始处显示，需要同时把CR和LF字符传给显示器或者打印机。但是，如果用汇编语言编程来实现这种操作就很麻烦。有时，在命令提示符下输入后，使得光标移开当前行，或者使用几条输出指令将所有输出都显示在一行，这时也可以不选用CR和（或者）LF。

使用较少的控制字符有“form feed”（$0C_{16}$），该字符使打印机退出某页；控制字符“horizontal tab”（09_{16}）在按下键盘的Tab键时生成；“backspace”（08_{16}）在按下键盘的Backspace键时生成；“delete”（$7F_{16}$）在按下键盘的Delete键时生成。注意：Delete键和Backspace键生成的代码不同。响铃（bell）字符（07_{16}）输出到显示器时会听见响铃声，有丰富编程经验的人员在真正需要响铃的时候才使用响铃字符。

许多大型计算机用“扩展二进制编码-十进制信息编码”（Extended Binary Coded Decimal Information Code，简称为EBCDIC，其发音为ib-SEE-dick或者eb-SEE-dick）。本书仅在讨论两种不同编码系统转换时会用EBCDIC编码作为例子。

练习1.2

1. 以下每个十六进制数可表示为一个十进制数或两个字符的ASCII值，请写出这两种表示。
   (a) 2A45　(b) 7352　(c) 2036　(d) 106E
2. 写出下列字符串的ASCII值，不要忘记空格和标点符号。回车和换行字符用CR和LF表示，如果写在一起CRLF（中间没有空格）表示回车换行功能。
   (a) January 1 is New Year's Day. CRLF
   (b) George said, "Ouch!"
   (c) R2D2 was C3P0's friend. CRLF["0" is the numeral zero]
   (d) Your name? [put two spaces after the question mark]
   (e) Enter value:[put two spaces after the colon]
3. 将下列ASCII序列输出到计算机显示器，会显示什么？
   (a) 62 6C 6F 6F 64 2C 20 73 77 65 61 74 20 61 6E 64 20 74 65 61 72 73

(b) 6E 61 6D 65 0D 0A 61 64 64 72 65 73 73 0D 0A 63 69 74 79 0D 0A

(c) 4A 75 6E 65 20 31 31 2C 20 31 39 34 37 0D 0A

(d) 24 33 38 39 2E 34 35

(e) 49 44 23 3A 20 20 31 32 33 2D 34 35 2D 36 37 38 39

1.3 有符号整数的二进制补码表示

本节详细探讨计算机中数的表示。前面已经介绍了两种表示数的方法，一种是用二进制表示（通常用十六进制表示），另一种是用ASCII码表示，但是这两种表示法有两个问题：（1）表示数的有效位是有限的；（2）如何表示负数并不明确。

第2章将讨论计算机硬件，现在应该知道计算机内存是以字节（Byte）为单位存储的，每一个字节包含8位（bit）[⊖]。假定内存中的数用ASCII码存储，一个ASCII码通常用一个字节存储。而ASCII码长度是7位，附加位（左侧或最高位）置0。为了解决上述表示法中的第二个问题，可在编码中包含一个负数符号。例如：用四个字符的ASCII编码来表示－817，就是2D，38，31和37。为解决第一个问题，通常用固定长度的若干字节数，位数不足时，在ASCII码的左边用0或者空格补充；也可以使用一个长度可变的字节数，但必须规定要表示的数的最后一个数位以ASCII码结束，也就是说用一个非数字字符结束。

计算机内部使用二进制表示数时，需选用某种固定长度位的表示方法。大多数中央处理单元（CPU）能进行变长的二进制数的运算。如Intel 80x86系列，可变长度有8位（1字节）、16位（1个字）[⊖]、32位（双字）和64位（四字）。

以697的字长二进制表示为例：

$$697_{10} = 1010111001_2 = 0000001010111001_2$$

二进制数表示的前面添加0是为了保证长度是16位，如果将该二进制数用十六进制数简化表示，即：

02	B9

这种表示法将贯穿全书，方框代表字节序列，每个字节的内容用十六进制表示，由于每个十六进制数用四位二进制数表示，因此，每个字节用两个十六进制数表示。如果用双字表示697，则前面需要用0补齐，即：

00	00	02	B9

前面介绍的数的表示法能够很好的表示非负数和无符号（unsigned）数，但不能表示负数。同时，任意给定长度都可表示一个最大无符号数，如字节长度，表示的最大无符号数为FF_{16}或者255_{16}。

二进制补码（2's complement）表示法与前面所讲的无符号数表示法很相似，但前者可以表示负数。二进制补码表示数时，应该明确其长度，所以可用“字长的二进制补码表示法”

⊖ 早期的计算机系统使用字节大小而不是8位。

⊖ 有些其他的计算机体系使用字大小而不是16位。