

全国高协组织教材研究与编写委员会审定

C 语言程序设计

徐庆生 编 著

科学普及出版社

·北京·

图书在版编目 (CIP) 数据

C 语言程序设计 / 徐庆生编著. —北京: 科学普及出版社, 2007.1

ISBN 978-7-110-06531-0

I. C... II. 徐... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2006) 第 142044 号

自 2006 年 4 月起本社图书封面均贴有防伪标志, 未贴防伪标志的为盗版图书。

科学普及出版社出版

北京市海淀区中关村南大街 16 号 邮政编码: 100081

电话: 010-62103210 传真: 010-62183872

科学普及出版社发行部发行

临沂市第二印刷厂印刷

*

开本: 787 毫米×1092 毫米 1/16 印张: 23.5 字数: 480 千字

2007 年 1 月第 1 版 2007 年 1 月第 1 次印刷

定价: 40.00 元

(凡购买本社的图书, 如有缺页、倒页、
脱页者, 本社发行部负责调换)

前 言

C 语言是一门应用广泛的计算机高级语言，不仅计算机专业学生需要学习 C 语言，而且许多非计算机专业的学生也在学习 C 语言，所以，C 语言似乎就成了深造计算机的必学语言。于是，有关 C 语言的各种水平考试也应应运而生。

然而，学习 C 语言究竟要达到什么目的，并不是所有学习者都明白的，我多次问过报名参加 C 语言等级考试的学生或在职人员，为什么要学习 C 语言？有各种回答，最常见的是：“因为开了这门课程”以及“为了通过 C 语言等级考试”。这些回答让我想到了许多学生学习英语的情况，从最初开始学习英语是为了能用英语交流，到后来是为了通过这门课程考试而学习英语。

多年教授 C 语言程序设计使我认识到，学生学习 C 语言只能有一个目的，那就是使用 C 语言，也就是编写 C 语言程序。教授 C 语言大致有两种方式，一种是以教授 C 语言本身为主，一种是以教授程序设计基础为主。目前，大多数教材属于满足前者要求的，这些教材以讲清 C 语言成分的正确使用为目标，虽然也涉及程序设计，但仅仅介绍程序设计基础知识，并没有从教学上系统地把这些程序设计基础知识与 C 语言组织成一个整体，特别是没有提供让学生循序渐进地体验使用 C 语言进行程序设计的“体系”。也许正是这样，此类教材在使用过程中，学生学习时感觉到要容易一些，更能较简单地“把握 C 语言”，也能较快地通过 C 语言的等级考试。毋庸置疑，这是这类教材的最大优点。

而满足后者的教材即满足以教授程序设计基础为主的教材相对要少，因为要实现这一目标的人恐怕要少得多了。这类教材虽然从教学上对程序设计方法进行了较为系统地组织，专业性较强，并且往往对 C 语言的组织又过于简单，不适合教授 C 语言的需要。所以，这类教材至少不适应于初学者，更不适应于非计算机专业的学生，甚至也不适应于计算机相关专业的低年级学生。

近几年来，作为楚雄师范学院院级重点课程建设项目的一项任务，我根据多年给计算机专业学生、非计算机专业学生和面向社会的培训班学员教授

C 语言的教学经验，编写了《C 语言程序设计》一书。本书首先按教学要求对 C 语言进行了系统的组织，也对程序设计基础进行了系统的组织，然后把它们有机地整合成一个体系。

本书特点

在编写本书时始终坚持的原则是：

(1) 给学习者提供一个循序渐进地体验使用 C 语言进行程序设计的过程的“场境”。

(2) C 语言的讲解要全面、准确和有效。既要保证语法上的严谨，也要保证有机地使它与程序设计结合起来。

(3) 尽可能地适应于专业学生、相关专业学生和非专业学生。要有适当取舍内容的余地，并且当舍去一部分有关程序设计的章节时不影响学习 C 语言。

本书还具有如下一些特点：

(1) 本书包含有较多的例子和实例，这些例子和实例都通过了 TC2.0 和 VC6.0 的测试，并且许多实例按照从简单情形到完整情形的顺序出现多次。例如求解一元方程 $ax^2+bx+c=0$ 的程序，它出现在第三章时，首先作为第 1 个例子默认 $a=0$ 即求解一元一次方程；然后在讲授了选择结构或 C 语言选择语句后，作为“自顶向下，逐步求精”一节的实例以更完整的情形再次出现；到了第四章讲授循环语句时再次出现该例子，且是按照一次运行可求解多个方程的需要来修改完善原有程序的；到了第五章讲授函数和模块化程序设计时，该实例又作为模块化的例子进行分析，给出了该实例的函数版本；到了第 15 章该实例再次作为菜单程序设计的例子被引入；最后到了第 16 章讲授文件处理时，按照事先把批量方程的系数存储在文件中的要求，该实例再次被引入。

这种处理方法有利于学生循序渐进地学习实例，掌握实用程序。

(2) 本书采用了国外教学上所谓的“解析法”对例子程序中每一句进行讲解，这样学生看例子程序时不用去查阅前面忘了的 C 语言内容，有利于学生课后看书和自学。

(3) 本书在描述算法时采用两种方式即自然语言和程序流程图，使问题变得较为简单。其中程序流程图虽然不被许多人看好，但作为理解算法，它描述算法的结构流程是比较清晰的。

(4) 每章开头都有本章学习的目标，结束时都有本章内容的小结，使得学生较容易把握住所学内容；每章根据教学目标还配有一定数量的练习题和上机实验题，这是学习 C 语言程序设计必不可少的；为了进一步使学习者更容易理解本书所讲授的内容，每章还配有一定数量的图示；同时，每章结束时都有一节专门介绍与本章内容相关的初学者容易出错的地方。

(5) 本书对较难的章节和习题，使用*号作了标识。

本书概览

本书包括 17 章和 4 个附录。第 1 章是学习 C 语言程序设计的入门，主要讲授机器语言、汇编语言和高级语言的概念及它们之间的关系，讲授简单 C 语言程序的结构以及 C 语言编程环境 VC6.0 的初步使用方法，即如何上机测试 C 语言程序。第 2 章讲授 C 语言的词法系统，主要包括 C 语言的运算符的使用、表达式以及编制简单 C 语言程序如数学公式计算程序的方法。第 3 章是选择控制结构，其讲授的主要内容包括 C 语言的选择控制语句、选择控制结构程序设计以及初步的结构化程序设计方法。第 4 章是重复控制结构，其讲授的主要内容包括 C 语言的三种循环控制语句、重复控制结构程序设计、算法的基本概念、性质、描述方法和算法好坏的概念，以及结构化程序的要求。第 5 章是函数，其讲授的主要内容包括 C 函数的概念、定义和使用、C 语言的作用域概念、C 递归函数的概念、模块化程序设计的基本思想和方法以及相关的一些概念介绍（如程序结构图、独立性原理和信息隐蔽等）。第 6 章是字符处理，其讲授的主要内容包括 C 语言字符的基本数据类型、字符处理的标准 C 函数以及程序验证的基本概念。第 7 章是基本数据类型，其讲授的主要内容包括 C 语言声明语句的作用、C 语言的基本数据类型、表达式类型转换的概念和方法。第 8 章是枚举类型和 typedef，其讲授的主要内容包括枚举类型及其应用、typedef 的应用以及多文件程序设计方法等。第 9 章是数组，其讲授的主要内容包括 C 语言一维数组和二维数组的基本概念、定义和使用、把数组传递给函数的方法以及简单的排序和查找程序等。第 10 章是指针，其讲授的主要内容包括指针的概念、定义和使用方法、指针与数组的关系以及指针在函数调用中传递参数的方法。第 11 章是字符串，其讲授的主要内容包括字符串的概念和存储方法、字符串的处理与标准的串处理和转换函数的使用以及把参数传递给 main 函数的方法等。第 12 章是存储类型与类型限定符，其讲授的主要内容包括存储类型的基本概念与类别以及 extern、

register、static 和 const 等类型限定符在程序中的使用方法。第 13 章是递归，其讲授的主要内容包括递归与迭代的概念以及分而治之的程序设计方法。第 14 章是结构与抽象数据类型，其讲授的主要内容包括结构的概念、定义和应用方法，联合的概念、定义和使用方法，位运算以及简单的线性链表的建立方法等。第 15 章是人机交互与菜单程序设计，其讲授的主要内容包括人机交互的概念和方式、菜单驱动程序设计方法以及简单的事件驱动程序设计概念。第 16 章是文件的处理，其讲授的主要内容包括文件的概念和操作原理、C 语言对文件的各种操作方法以及相应的文件程序设计等。第 17 章是 C 语言的其他专题，其讲授的主要内容包括 C 预处理命令及其使用、C 语言的变长参数的使用方法、C 的内联函数的概念、C 的动态内存分配的方法以及 goto 语句等。4 个附录分别提供了 ASCII 码字符集、C 运算符的优先性和结合性、函数 printf()和 scanf()详解与 C 语言常用标准函数等。

本书适用范围及学习建议

本书可以作为计算机专业及有关专业的《高级语言程序设计》、《C 语言程序设计》、《C 程序设计语言》以及《程序设计基础》等课程的教材，也适合自学者使用。在使用时，不同的学习者可以进行适当的取舍。下面是一个供参考的建议。

1) 如果学习 C 语言的目的是理解或掌握程序设计基础，则一般情况可以学习所有不带*号的章节。但如果是专科或专科以下学生，或者是作为第一门语言的自学者，则可以不学下列章节：

第三章之 3.7；第四章之 4.7；第五章之 5.6，5.9；第六章之 6.5；第八章之 8.3；第九章之 9.6；第十三章；第十四章之 14.5，14.6，14.7，14.10；第十五章；第十六章之 16.7；第十七章。

2) 如果学习 C 语言的目的是理解或掌握语言本身，则可以不学下列章节：

第三章之 3.7；第四章之 4.5，4.7；第五章之 5.6，5.9；第六章之 6.5；第八章之 8.3；第九章之 9.4，9.5，9.6；第十一章之 11.5，11.7；第十三章；第十四章之 14.5，14.6，14.7，14.10；第十五章；第十六章之 16.7；第十七章。

徐庆生

目 录

第1章 入门	1
1.1 引言	1
1.2 C语言	9
1.3 C编译系统的使用	20
1.4 小结	25
1.5 练习题	27
1.6 上机实验题	29
第2章 词法元素、运算符和C系统	31
2.1 词法元素	32
2.2 运算符与表达式	36
2.3 例子	38
2.4 C系统	40
2.5 常见编程错误	44
2.6 小结	45
2.7 练习题	47
2.8 上机实验题	51
第3章 选择控制结构	52
3.1 引言	52
3.2 条件表达式	54
3.3 if选择结构	57
3.4 if_else选择结构	58
3.5 switch多情形选择结构	67
3.6 嵌套选择结构	70
3.7 自顶向下，逐步求精	72
3.8 常见编程错误	78

3.9 小结	79
3.10 练习题	81
3.11 上机实验题	85
第4章 重复控制结构	86
4.1 while 语句	86
4.2 for 语句	92
4.3 do 语句	93
4.4 break 和 continue 语句	94
4.5 问题求解：求小于 n 的所有素数	97
4.6 复杂程序控制	99
4.7 算法*	101
4.8 常见编程错误	103
4.9 小结	103
4.10 练习题	105
4.11 上机实验题	107
第5章 函数	108
5.1 函数的使用	108
5.2 函数的定义	110
5.3 函数的返回	112
5.4 函数的声明	114
5.5 问题求解	114
5.6 结构化程序设计	117
5.7 作用域	121
5.8 递归	124
5.9 程序结构图*	125
5.10 常见编程错误	126
5.11 小结	126
5.12 练习题	129
5.13 上机实验题	132

第6章 字符处理	133
6.1 数据类型 char	133
6.2 getchar()和 putchar()用法	134
6.3 ctype.h 中的宏	136
6.4 问题求解	137
6.5 程序验证: asert()宏的用法*	139
6.6 系统问题	140
6.7 常见编程错误	140
6.8 小结	140
6.9 练习题	142
6.10 上机实验题	144
第7章 基本数据类型	145
7.1 声明和表达式	145
7.2 基本数据类型	146
7.3 字符和数据类型 char	147
7.4 int、short、long 和 unsigned	148
7.5 浮点类型	149
7.6 sizeof()运算符	150
7.7 数学函数	150
7.8 类型转换	152
7.9 问题求解	153
7.10 常见编程错误	155
7.11 小结	156
7.12 练习题	158
7.13 上机实验题	159
第8章 枚举类型和 typedef	160
8.1 枚举类型	160
8.2 typedef 的用法	162
8.3 例子: 石头、剪刀、布	162
8.4 常见编程错误	169

8.5 系统考虑	169
8.6 小结	169
8.7 练习题	170
8.8 上机实验题	171
第9章 数组	173
9.1 数组	173
9.2 数组使用举例	174
9.3 把数组传给函数	177
9.4 排序	179
9.5 查找	180
9.6 问题求解	182
9.7 二维数组	186
9.8 常见编程错误	188
9.9 小结	189
9.10 练习题	189
9.11 上机实验题	190
第10章 指针	191
10.1 指针	192
10.2 指针与函数引用调用	193
10.3 指针与数组	196
10.4 问题求解	197
10.5 常见编程错误	199
10.6 小结	199
10.7 练习题	200
10.8 上机实验题	202
第11章 字符串	203
11.1 字符串	203
11.2 例子	204
11.3 用指针处理串	206
11.4 标准库中串处理函数	209

11.5	把参数传给 main()	210
11.6	字符串转换函数	211
11.7	问题求解	212
11.8	常见编程错误	213
11.9	小结	214
11.10	练习题	215
11.11	上机实验题	215
第 12 章	存储类型与类型限定符	216
12.1	存储类型	216
12.2	存储类型 extern	217
12.3	存储类型 register	218
12.4	存储类型 static	219
12.5	存储类型 const	220
12.6	小结	223
12.7	练习题	224
第 13 章	递归	227
13.1	递归	227
13.2	递归与迭代	232
13.3	分而治之的方法	233
13.4	问题求解：梵塔	235
13.5	小结	237
13.6	练习题	237
13.7	上机实验题	238
第 14 章	结构与抽象数据类型	239
14.1	结构	239
14.2	结构与函数	242
14.3	在结构声明中使用 typedef	244
14.4	结构数组与结构指针	245
14.5	问题求解：学生记录	248
14.6	自引用指针	252

14.7	线性链表	253
14.8	联合	257
14.9	位运算与位字段	259
14.10	问题求解	261
14.11	常见编程错误	265
14.12	小结	265
14.13	练习题	267
14.14	上机实验题	268
第 15 章	人机交互与菜单驱动程序设计	269
15.1	界面与人机交互	269
15.2	菜单设计	270
15.3	问题求解	277
15.4	事件驱动*	283
15.5	信号处理*	285
15.6	问题求解*	287
15.7	小结	289
15.8	练习题	290
15.9	上机实验题	291
第 16 章	文件处理	292
16.1	文件和流	294
16.2	建立顺序文件	295
16.3	顺序文件的读取	300
16.4	顺序文件的修改	303
16.5	建立随机存取文件	306
16.6	随机文件存取	309
16.7	例子：学生成绩管理	313
16.8	常见编程错误	320
16.9	小结	321
16.10	练习题	324
16.11	上机实验题	325

第 17 章 C 语言的其他专题*	326
17.1 预处理	326
17.2 变长参数与 inline	332
17.3 动态内存分配: 函数 calloc 和 realloc	335
17.4 无条件转移: goto 语句	337
17.5 练习题	338
17.6 上机实验题	339
附 录	340
A ASC II 字符集	340
B 运算符的优先级与结合性	341
C 函数 printf()与 scanf()详解	342
D C 语言常用函数	350
参考资料	361

第 1 章 入 门

学习目标

- (1) 了解计算机的组成及运行原理，初步理解简单机器语言程序的执行特征。
- (2) 理解汇编语言指令与机器语言指令的关系，理解高级语言与汇编语言的关系，理解编译程序或编译器的功能与作用，初步理解掌握执行程序的过程。
- (3) 了解高级语言发展简史。
- (4) 初步理解算法和程序的概念，掌握简单 C 程序的基本结构、理解 C 程序设计基本过程及其概念，能初步使用格式化输入输出语句。
- (5) 能在 VC 编程环境中输入程序，并使之运行。

1.1 引 言

C 语言是一门相对难以掌握的程序设计语言，通常只有富有经验的程序员或计算机及其相关专业的学生才去学习它，事实上，它是许多学校计算机及其相关专业学生的一门必修课程。通过对它的学习，学生将熟悉程序设计基本知识和基本技能，并积累编程经验，而这正是高级语言程序设计主要目的之一。本章是学习 C 语言程序设计的一个简短入门。

1.1.1 计算机是什么

我们操作过计算机的人都知道，计算机就是一种设备，它能让我们进行文字处理、绘制图形、玩游戏、听音乐、制作动画、上网查询信息和观看网络电影等等。这是从计算机应用角度上的认识，实际上，计算机是能够以比人快几百万甚至几十亿倍的速度执行计算和逻辑判断的设备。这种说法可能让人感到有点神奇甚至迷惑，“文字处理、绘制图形、玩游戏、制作动画、听音乐、上网查询信息和观看网络电影等”是怎样与“执行计算和逻辑判断”联系起来呢？

实际上，计算机系统包含两大部分，一是**硬件**本身，它包括键盘、鼠标、显示器、磁盘和主机（箱）等；另一是**软件**。计算机硬件本身提供了执行计算和逻辑判断的功能，这些功能表现为一组**指令**，如加法指令、减法指令、取数指令、存数指令、输入指令和输出指令以及逻辑判断类指令等；而软件是程序员利用计算机指令集，规划组织完成令我们激动的玩游戏、听音乐和制作动画等计算机应用的特定指令序列。软件也可简称为**程序**。实际上，软件与程序是不等价的，简单说，软件=程序+相关文档资料。

计算机硬件是在程序的指令集控制下处理数据的。计算机程序控制着计算机，使它按顺序执行一系列动作，这些动作是由程序员规划的，并用指令描述的。

1.1.2 计算机的组成及运行

各种计算机无论外观差别有多大，每一台计算机实际上都可划分为五个逻辑单元（或称五大部分），如图 1-1 所示。

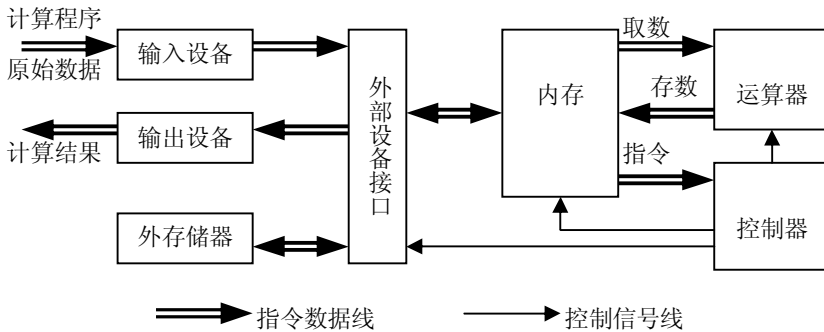


图 1-1 计算机的基本组成

(1) **输入单元**。输入单元是计算机接收信息的部分。它从各种输入设备读取信息（数据和计算机程序），并把这些放置到其他处理信息的单元中。当今计算机的大多数信息是通过键盘输入的，但也逐渐出现使用其他许多输入设备来进行信息输入，如扫描仪、声音输入设备等。

(2) **输出单元**。输出单元是计算机输出信息的部分。它把计算机处理过的信息放置到各种输出设备中，从而使这些信息能够被计算机外部使用。当今计算机的大多数信息是通过屏幕显示和纸张打印输出的，同时也出现使用其他许多输出设备来进行信息输出，如音箱等。

(3) **内存单元**。内存单元是计算机中存取速度快、容量相对较少的储存部分。它能够记忆来自输入单元的信息，因而能够在需要的时候立即处理这些信息。内存单元还能够记忆被处理过的信息，直到输出单元把信息放到输出设备上。内存单元经常被称为**内存或主存**。除“只读存储器（ROM）”外，内存在计算机电源关闭后将不起作用。

(4) **中央处理单元（CPU）**。中央处理单元是计算机中的“加工”和“管理”部分。它负责执行各种计算如加、减、乘、除等，负责逻辑判断，如比较内存中的两个数是否相等，这部分又称为**运算器**。同时，它还是计算机的协调者，负责管理对其他部分的操作，这部分又称为**控制器**。在应该把信息读到内存单元中时，CPU 就会向输入设备发出请求；当要把内存中信息用于计算时，CPU 会把这种请求通知其内部的运算器；当要把内存中的信息发送给某个输出设备时，CPU 就会告诉输出设备。

(5) **二级存储单元**。二级存储单元是计算机中可以长久保存信息的高容量存储部分。没有被其他单元使用的程序和数据通常放在二级存储单元中（如磁盘），这些信息可能需要几个小时、几天、几个月甚至几年才会被使用。二级存储单元也称为**外存储器，外存或称为辅存**。

计算机运行的过程，就是顺次执行程序中指令的过程。计算机的指令以及运行时数据都是采用二进制数表示的。下面举一个简单的例子说明计算机的运行。

例子程序运行后，接受用户输入的两个数，计算其平均值，并输出结果。例子程序

的执行过程为：

- (1) 输入两个数存放于变量 a 和 b 中；
- (2) 计算(a+b)/2，并将其结果存储于变量 X 中；
- (3) 输出 X 中的值；

当用计算机的指令来完成上述过程时，需要考虑到计算机的具体特性，如计算机中没有变量，而是用**存储单元**来表示等。完成上述过程的指令序列及其运行时在内存的存储布局如图 1-2 所示。

内存单元地址	内存
00000100	(表示 a)
00000101	(表示 b)
00000110	(表示 X)
00000111	10000110 01100000
00001001	10100010 00000100 00000000
00001100	10000110 01100000
00001110	10100010 00000101 00000000
00010001	10100000 00000100 00000000
00010100	00000000 11000000 00000101 00000000
00011000	10110001 00000010
00011010	11110110 11110001
00011100	10100010 00000110 00000000
00011111	10111010 11010101 00000011
00100010	11101110
00100011	11110100

图 1-2 计算 $X=(a+b)/2$ 指令序列

每 8 位二进制数称为一个字节(Byte)，每一条指令用若干个字节的二进制代码表示，并存储到内存单元中。最小内存单元为 1 个字节，它必须用一个编号来表示，这个编号称为**单元地址**。在图 1-2 中我们并没有把每个单元的地址标出来，只标出每条指令代码存储的起始单元的地址。图 1-2 中的开始三个字节存储单元分别表示变量 a、b 和 X，用于存储它们的值。其他存储单元用于存储指令，这些指令的含义解释如下。

第一条指令用 2 个字节数据“10000110 01100000”表示 CPU 从输入设备读进一个数 (a) 并放置到寄存器 AL 中。

第二条指令用 3 个字节数据“10100010 00000100 00000000”表示 CPU 把寄存器 AL 中数放置到内存中地址编号为 00000100 的存储单元 (1 个字节) 中保存。其中第 3 个字节数据和第 2 个字节数据合成一个 16 位二进制数据 (即 00000000 00000100)，表示

存储单元 (a) 的地址。

第三条指令用 2 个字节数据“10001110 01100000”表示 CPU 从输入设备读进一个数 (b) 并放置到寄存器 AL 中；

第四条指令用 3 个字节数据“10100010 00000101 00000000”表示 CPU 把寄存器 AL 中数放置到内存中地址编号为 00000101 的存储单元 (1 个字节) 中保存。其中第 3 个字节数据和第 2 个字节数据合成一个 16 位二进制数据 (即 00000000 00000101)，表示存储单元 (b) 的地址。

第五条指令用 3 个字节数据“10100000 00000100 00000000”表示 CPU 从内存中地址编号为 00000100 的存储单元读进 1 个字节数据 (a) 并放置到 AL 寄存器中。

第六条指令用 4 个字节数据“00000000 11000000 00000101 00000000”表示 CPU 把内存中地址编号为 00000101 存储单元的数据 (b) 加到 AL 寄存器中。

第七条指令用 2 个字节数据“10110001 00000010”表示 CPU 把数 00000010 即 2 存储到 CL 寄存器中。

第八条指令用 2 个字节数据“11110110 11110001”表示 CPU 用 CL 中数去除 AL 寄存器中数，其结果存储在 AL 寄存器中。

第九条指令用 3 个字节数据“10100010 00000110 00000000”表示 CPU 把寄存器 AL 中数传送到内存中地址编号为 00000110 的存储单元 (X) 中保存；

第十条指令用 3 个字节数据“10111010 11010101 00000011”表示 CPU 把数 00000011 11010101 即 03D5H (十六进制) 存储到 DX 寄存器中；03D5H 表示显示器的端口号。

第十一条指令用 1 个字节数据“11101110”表示 CPU 把 AL 寄存器中数送到 DX 寄存器所指向的端口连接的设备即显示器。

第十二条指令用 1 个字节数据“11110100”表示 CPU 停止执行指令。

计算机的指令集也称为**机器语言**，上述的指令序列也称为**机器语言程序**。注意，该例子中用汉语描述计算过程时的每一步，在用机器语言描述时需要多条指令。

1.1.3 机器语言、汇编语言和高级语言

机器语言就是计算机 CPU 的指令集合，用它表示的计算程序即机器语言程序可以直接由计算机执行。我们所谓的使用计算机，其实质就是通过使用机器语言来完成计算任务，所以从这个角度上讲，机器语言就是机器的“代表”或“化身”。

但机器语言程序难理解，因为机器指令是用二进制数据表示的。为了便于记忆、理解机器指令或机器语言程序，人们用字母符号来表示机器指令，这种字母符号一般用一个有意义单词的前几个字母组成，如 ADD 表示加法指令，MOV 表示传送数据等等。符号化的指令称为**汇编指令**，用它表示的程序称为**汇编语言程序**。

上节例子中的机器语言程序对应的汇编语言程序为下表右部分所示：