

精通

C 程序设计教程

赵宏宇 编著



西南交通大学出版社

精通 C 程序设计教程

赵宏宇 编著

西南交通大学出版社

· 成都 ·

图书在版编目 (C I P) 数据

精通 C 程序设计教程 / 赵宏宇编著. — 成都: 西南交通大学出版社, 2004.2

ISBN 7-81057-807-3

I. 精... II. 赵... III. C 语言-程序设计-高等学校-教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 113288 号

精通 C 程序设计教程

赵宏宇 编著

*

责任编辑 张 波

封面设计 何东琳设计工作室

西南交通大学出版社出版发行

(成都二环路北一段 111 号 邮政编码: 610031 发行部电话: 87600564)

<http://press.swjtu.edu.cn>

E-mail: cbsxx@swjtu.edu.cn

四川森林印务有限责任公司印刷

*

开本: 787 mm × 1092 mm 1/16 印张: 22.5

字数: 533 千字

2004 年 2 月第 1 版 2004 年 2 月第 1 次印刷

ISBN 7-81057-807-3/TP · 325

定价: 28.00 元

序

以程序设计为中心的计算机技术正在世界范围内日新月异地发展着，对于从事开发计算机系统软件或者编写应用程序的各类专业人士来说，掌握和精通计算机高级语言程序设计是一件不可或缺的看家本领。

在计算机不同发展阶段中，曾出现过多种多样的计算机语言，其中 C 语言以其功能齐全、使用方便，既有高级语言的简洁通用形式，又有低级语言的高效灵活特性，因而大受青睐并得到持久地推广使用。然而，由于 C 语言本身的难度和优秀教材的匮乏，很多曾学习过 C 语言的人员常常感到课堂中学习过的算法语言知识难以实用，往往需要再次深入学习才能胜任 C/C++ 的开发工作。

本书作者长期从事 C/C++ 程序设计方面的教学与科研工作，深知“授人以鱼不如授人以渔”的哲理，对 C 语言如何才能由陌生到熟悉，由熟悉到精通有切身体会，因而在本书的编写中，特别重视学习者在算法设计和实际编程能力方面的培养训练。书中的很多内容，比如，运算符与表达式、指向数组的指针、函数指针、变量的属性、基本程序设计技术、递归程序设计与分析、动态数据结构、文件操作等章节的合理安排，以及万年历的打印、排列与组合的产生、多种随机数等问题的解法等简洁编程算法，都是国内一般 C/C++ 程序设计教材中难以见到的。阅读之余，颇有令人耳目一新、茅塞顿开之感。

本书在内容取舍和章节安排中凝聚了作者丰富的教学经验和科研积累，力求概念清晰、循序渐进、例题丰富、信息量大。读者如能系统地学习本书内容，并配合以大量上机编程实践，一定会对将来从事软件开发或者在计算机程序设计方面进一步的深造产生积极而深远的促进作用。

靳蕃

2004 年 1 月

前 言

本教程对 C 语言以及程序设计的基础知识进行了全面介绍。教材围绕程序设计的语法和算法两大核心主题展开教学内容，在体系结构与教学思想上突出了语法学习和算法实践并重的鲜明特点。作者的目标是使本教材对读者易学，对教师好用，教学内容的全面和深刻能够对读者将来从事应用软件开发、后续学习数据结构等课程以及报考研究生有很高的实用价值。

C 语言是一门使用广泛而又颇有学习难度的程序设计语言，因为它的语言内容比其他高级语言（比如 FORTRAN、PASCAL 和 BASIC）具有更丰富的语法和更灵活的使用形式。本教材以 ANSI C88（有文献称为 ANSI C89）标准为基础，全面覆盖了 C 语言的各种语法，特别是对 C 语言特有的比较复杂的各种语法现象进行了深刻总结和剖析，目的是使读者一步一步地从 C 语言入门走向精通。考虑到国内大学目前所使用的很多 C 语言教材所介绍的 C 语言以及全国计算机等级考试中涉及的个别语法现象使用的仍然是 ANSI C83 标准，本教材对这些标准的变化进行了必要说明。C 语言的最新标准是 ISO C99，但是，到 2003 年底为止，绝大多数实际应用的 C 和 C++ 语言系统尚未支持 ISO C99 标准。

本教程特别重视对学生算法设计和实际编程能力的培养，程序例题丰富，并面向实践应用，全部例题都通过了 Turbo C2.0、Borland C++3.1 和 Visual C++6.0 系统的测试。各种例题围绕语法应用和算法设计主题进行了系统分类，融入了作者多年从事 C/C++ 软件开发和程序设计课程教学的经验。算法程序内容的丰富性、新颖性和实用性相信能够使读者快速走向深入。

教材立足于高中毕业水平，教学内容完全覆盖了全国计算机等级考试（二级）大纲要求，内容体系结构与章节安排符合循序渐进的学习特点，有利于读者自学。教材在文字讲解上力求通俗易懂、细致深刻而又言简意赅。各章附有经过精心安排的习题，包括部分等级考试题型，有助于读者掌握教学内容和应试。为帮助学生理解和掌握程序以及编程方法，很多例题都附带了“问题与思考”，部分“问题与思考”在教材中直接给出了答案，而更多的问题与思考留给读者作为练习。

为了与 C++ 很好地衔接，教材讲述的 C 语言考虑了 C++ 适应性，并以读者将来会学习 C++ 为背景。

全书共分为 12 章，第 1 章介绍 C 程序设计的基础知识，这些基础知识充分考虑了读者可能没有任何程序设计知识的情况。第 2 章介绍 C 语言的基本语法，包括数据类型、变量、常量、表达式与简单 C 程序设计。第 3 章介绍 C 语言的各种控制语句，即选择结构与循环结构。第 4 章分类介绍常用的基础性程序设计技术，包括打印字符图形、求最大值与最小值问题、统计（计数）问题、递推迭代问题、搜索求解问题以及与数论有关的一些算法。

第5章介绍C语言的数组。第6章和第9章介绍C语言的指针，分别称为指针初步和指针深入。在第6章指针初步中，还专门介绍了C语言的字符串处理。第9章指针深入中包括多级指针、指针数组、指向数组的指针和函数指针。本教材将指针分为两个部分进行介绍也是基于循序渐进学习规律的考虑。第7章对C语言的函数进行了全面讨论，包括函数的定义与使用、变量（标识符）的属性与作用域、内部函数与外部函数、多个源程序文件编程等内容。第7章还包含常用的排序算法（选择排序、插入排序和冒泡排序）、查找算法（顺序查找与折半查找）以及递归程序设计与分析。第8章介绍C语言的编译预处理，包括宏定义、包含文件以及条件编译。第10章对枚举类型、结构体、联合体、位段以及类型定义进行介绍，还包含链表与二叉树。教材的第11章对C语言的文件操作进行全面介绍并给出了比较丰富的字符文件与二进制文件应用例程。第12章对程序设计入门课程中偏难的问题、一些趣味性较强的问题、数值计算问题以及产生随机数问题进行了介绍，目的是开拓读者的视野，启发读者的思维。其中，数值计算以及产生随机数等问题被安排在最后一章，是考虑到这些问题涉及一些高等数学、线性代数和概率论的知识，而程序设计课程一般是安排在大学第一学期或者第二学期进行教学的，因此这些知识对于刚进校的一年级大学生来说，需要一段时间才能学习和掌握。

本书可以用做大学本科、专科各专业计算机系列课程之程序设计课程教材，全国计算机等级考试（二级C程序设计）教材，以及广大工程技术人员，编程爱好者的参考书。

教师在使用本教材时，可以根据学生的实际情况与教学要求（比如，是本科还是专科，是计算机专业还是非计算机专业）安排教学进度和内容。使用本教材的理论授课学时数建议为48~64学时，每周上机时间建议不低于3小时。参考教学计划如下表所示。

参 考 教 学 计 划

教 学 内 容	48 学时本科教学参考计划		
	授课学时	授课内容	上机学时
第1章 C 程序设计基础知识	3	全部（含补码不含浮点数）	0
第2章 C 源程序基本语法与简单顺序结构程序设计	5	全 部	3
第3章 选择结构与循环结构	4	全 部	3
第4章 基本程序设计技术	5	全 部	9
第5章 数 组	2	全 部	3
第6章 指针初步	4	全部（含字符串处理）	6
第7章 函 数	8	全部（含排序/查找/递归）	6
第8章 编译预处理	2	全 部	3
第9章 指针深入	2	全 部	3
第10章 派生数据类型	8	全部（含链表不含二叉树）	6
第11章 文件操作	5	全 部	6
第12章 算法与 C 程序综合举例	自学为主	穿插选讲	自拟

由于本教材具备语法深入全面与算法实践并重的特色，相信各层次的读者，比如，从未学习过程序设计的、学习过其他程序设计知识而希望学习 C 语言的、甚至已经学习过 C 语言和 C++ 语言的读者，都能从本教材中汲取营养。

最后，作者要感谢西南交通大学的尹治本教授、张翠芳教授、周荣辉教授，他们给予了作者极大的鼓励并为本书的成稿提出了许多宝贵的建议。此外，听过作者讲课的广大学生也给予了作者非常热情的支持。

当然，作者也希望广大读者在使用本教材的过程中提出宝贵意见并对书中的错误和疏漏提出批评指正。

本教材的写作是在作者使用多年的电子讲稿基础上完成的。与本书配套的视频课件和电子讲稿即将由西南交通大学音像出版社正式出版发行。

作 者

2003 年 12 月 28 日于成都

目 录

第 1 章 C 程序设计基础知识

1.1 程序语言与编程方法的发展	1
1.1.1 机器语言	2
1.1.2 汇编语言	3
1.1.3 高级语言	4
1.1.4 程序语言集成开发工具	6
1.1.5 程序语言的分代与编程方法的演变	7
1.1.6 指令、程序与软件	9
1.2 C 语言的历史发展与特点	10
1.2.1 C 语言的历史发展	10
1.2.2 C 语言的特点	11
1.3 计算机中的信息存储	13
1.3.1 二进制、八进制、十六进制	13
1.3.2 整数与实数的存储格式	16
1.3.3 字符编码	22
1.3.4 磁盘文件	23
1.4 语法、数据结构与算法	24
1.4.1 语法	24
1.4.2 数据结构	26
1.4.3 算法	26
习 题	31

第 2 章 C 源程序基本语法与顺序结构简单程序设计

2.1 C 语源程序的组成与结构	33
2.1.1 C 源程序的语义字符	34
2.1.2 C 语言的标识符与关键字	34
2.1.3 C 语言的句法单位	36
2.1.4 C 语言的函数	39
2.1.5 C 源程序的一般结构	39

2.1.6 C 源程序风格与书写规范	40
2.2 调试第一个 C 程序	40
2.2.1 Windows 的“MS-DOS 方式”	41
2.2.2 用 Turbo C2.0 调试 C 程序	43
2.2.3 用 Turbo C++3.0 或 Borland C++3.1 调试 C 程序	47
2.2.4 用 Visual C++6.0 调试 C 程序	48
2.3 C 语言的基本数据类型、常量与变量	50
2.3.1 C 语言的基本数据类型	50
2.3.2 C 语言的常量	53
2.3.3 C 语言的变量	57
2.4 C 语言的运算符与表达式	59
2.4.1 算术运算符、数学库函数与数学表达式	59
2.4.2 二进制位运算符与位运算表达式	64
2.4.3 关系、逻辑运算符及关系、逻辑表达式	68
2.4.4 类型强制运算符与求存储长度运算符	70
2.4.5 赋值运算符与赋值表达式	71
2.4.6 条件运算符和逗号运算符	72
2.4.7 运算符的优先级与结合性	73
2.4.8 表达式数据类型的计算转换规则	73
2.4.9 表达式应用举例	75
2.5 C 程序的输入与输出	76
2.5.1 单个字符的输入与输出函数	76
2.5.2 格式化输出函数 printf	78
2.5.3 格式化输入函数 scanf	82
2.6 顺序结构简单程序设计	86
习 题	88

第 3 章 选择结构与循环结构

3.1 选择结构	91
3.1.1 if 语句	91
3.1.2 switch 语句	97
3.2 循环结构	99
3.2.1 while 循环语句	100
3.2.2 do...while 循环语句	103
3.2.3 for 循环语句	104
3.2.4 多重循环	106
3.2.5 循环体中的 break 语句与 continue 语句	107

3.3 goto 语句	110
习 题	110

第 4 章 基本程序设计技术

4.1 计数(统计)问题	114
4.2 求最大值与最小值	116
4.3 递推迭代(数列求和与求积)	118
4.4 字符图形	125
4.5 搜索(穷举)求解	129
4.6 数论有关问题	131
习 题	135

第 5 章 数 组

5.1 一维数组	138
5.1.1 一维数组的定义	138
5.1.2 访问一维数组的元素	138
5.1.3 一维数组的初始化	139
5.2 二维数组	140
5.2.1 二维数组的定义	140
5.2.2 访问二维数组的元素	141
5.2.3 二维数组的初始化	141
5.3 多维数组	144
5.3.1 多维数组的定义与元素访问	144
5.3.2 多维数组的初始化	145
习 题	146

第 6 章 指针初步

6.1 指针的概念与基本操作	148
6.1.1 指针的概念	148
6.1.2 指针常量与变量	149
6.1.3 指针的基本运算与操作	151
6.2 指针与一维数组	155
6.2.1 一维数组名	155
6.2.2 元素等价形式与地址等价形式	156
6.2.3 一维动态数组	158

6.3 字符串处理	160
6.3.1 字符串的存储格式	160
6.3.2 程序中存储字符串与访问字符串的方法	161
6.3.3 字符串的输入与输出	162
6.3.4 字符串操作常用库函数	162
6.3.5 字符串处理程序设计举例	164
6.4 数组整体拷贝	167
习 题	168

第 7 章 函 数

7.1 函数的定义与调用	170
7.1.1 定义函数的语法格式	170
7.1.2 函数调用	173
7.2 函数的参数传递	176
7.2.1 实参表达式向形参变量传值	176
7.2.2 指针变量作形参	177
7.2.3 一维数组作形参	180
7.2.4 二维数组作形参	182
7.2.5 多维数组作形参	183
7.3 变量的特性	184
7.3.1 全局变量	184
7.3.2 局部变量	185
7.3.3 变量的存储类型	186
7.3.4 变量的生存期与变量的初始化	187
7.3.5 全局（外部）变量说明	189
7.4 外部函数与内部函数	190
7.4.1 外部函数与内部函数的概念	190
7.4.2 外部函数说明	190
7.4.3 工程文件	191
7.5 返回值为指针类型的函数	192
7.6 排序与查找	195
7.6.1 选择排序	195
7.6.2 插入排序	196
7.6.3 冒泡排序	198
7.6.4 索引排序	200
7.6.5 顺序查找	201
7.6.6 折半查找	202

7.7 递 归	203
7.7.1 递归程序设计	203
7.7.2 递归程序分析	207
习 题	211

第 8 章 编译预处理

8.1 宏 定 义	216
8.1.1 不带参数的宏定义	216
8.1.2 带参数的宏定义	217
8.1.3 宏名字的作用域	218
8.2 包含文件	219
8.3 条件编译	220
习 题	222

第 9 章 指针深入

9.1 多级指针与指针数组	223
9.1.1 多级指针	223
9.1.2 指针数组	224
9.2 指向数组的指针	226
9.2.1 定义指向 m 维数组的指针变量	227
9.2.2 指向数组的指针类型名、类型强制与指针运算	227
9.2.3 m 维数组与指向数组的指针	230
9.2.4 m 维动态数组	233
9.2.5 指向数组的指针作函数的形参	235
9.2.6 指向数组的指针作函数的返回值	235
9.2.7 指向数组的指针作数组元素	236
9.3 函数指针	236
9.3.1 定义函数指针变量	236
9.3.2 函数指针变量的初始化、赋值操作与函数调用	237
9.3.3 函数指针作函数的形参	238
9.3.4 函数指针作函数的返回值	238
9.3.5 函数指针作数组元素	239
9.4 带参数的 main 函数	240
习 题	241

第 10 章 派生数据类型

10.1 枚举类型	243
10.2 结 构 体	245
10.2.1 定义结构体	246
10.2.2 结构体的成员变量	247
10.2.3 结构体成员变量的输入与输出	248
10.2.4 结构体类型作函数形参与返回值	250
10.3 联 合 体	251
10.4 位 段	252
10.4.1 位段的定义与存储结构	252
10.4.2 位段型成员变量的使用	253
10.5 类型定义	254
10.6 链 表	255
10.6.1 链表的图形表示与结点数据结构	255
10.6.2 链表的基本操作	257
10.6.3 链表算法程序举例	262
10.6.4 静态链表	270
10.7 二 叉 树	271
10.7.1 二叉树的基本概念	271
10.7.2 二叉树结点结构与数据类型	271
10.7.3 遍历二叉树	272
10.7.4 建立二叉树	274
10.7.5 删除二叉树	274
10.7.6 二叉树算法程序举例	275
习 题	279

第 11 章 文件操作

11.1 C 语言文件操作基本知识	283
11.2 C 语言文件操作数据结构	284
11.3 C 语言文件操作函数	284
11.3.1 文件的打开与关闭函数	285
11.3.2 文件读写函数	286
11.3.3 文件的读写位置指针与操作函数	288
11.3.4 字符文件操作与二进制文件操作的区别	290
11.3.5 操作系统文件操作命令函数	290
11.4 字符文件应用	291

11.5 二进制文件应用	295
习 题	297

第 12 章 算法与 C 程序综合举例

12.1 约瑟夫问题	301
12.2 打印万年历	302
12.3 排列问题与组合问题	306
12.3.1 排列问题	306
12.3.2 组合问题	311
12.4 汉诺塔问题	313
12.5 八皇后问题	315
12.6 数值计算方法举例	316
12.6.1 用辛普森法求定积分	316
12.6.2 用高斯消元法解线性方程组	318
12.6.3 用高斯-塞德尔 (Gauss-Seidel) 迭代法求解线性方程组	322
12.6.4 用二分法求方程的根	324
12.6.5 用截弦法求方程的根	325
12.6.6 用牛顿迭代法求方程的根	326
12.7 产生随机数	327
12.7.1 均匀分布随机数	327
12.7.2 正态分布随机数	329
12.7.3 指数分布随机数	330
12.7.4 泊松分布随机数	330
习 题	331
附录 I 标准 ASCII 码表	334
附录 II C 语言常用库函数	335
附录 III C 语言向 C++ 语言过渡	339
附录 IV C++ 的 new 和 delete 操作符	341
参考文献	343

第1章 C 程序设计基础知识

1.1 程序语言与编程方法的发展

1946 年在美国陆军部的主持下，美国宾夕法尼亚大学研制成功了世界上第一台电子计算机 ENIAC (Electronic Numerical Integrator And Calculator)，即“电子数字积分计算器”。

ENIAC 由 18000 多个电子管和 1500 多个继电器等原件组成，重达 30 t，占用了每层面积为 200 m² 的两层楼，每秒钟只能做 5000 次加法或 400 次乘法运算。

1946 年 6 月，世界著名的数学家美籍匈牙利人冯·诺依曼 (Von.Neumann) 发表了《关于电子计算机逻辑设计的初步探讨》的报告，提出了用二进制存储程序的新思想，即计算机由运算器、控制器、存储器、输入设备与输出设备 5 大基本硬件构成，见图 1.1。其基本工作原理是：程序与数据保存在存储器中；控制器从存储器中顺序取出指令，经过解析后执行指令；运算所需要的数据包含在程序指令流中；运算结果保存在存储器中；输入设备将程序与数据翻译成二进制形式并存入存储器；而输出设备则将存储器中的二进制数据转换为人能够直观接受的形式。根据冯·诺依曼提出的思想，1949 年在英国剑桥大学完成了第一台存储程序的计算机 EDVAC (Electronic Discrete Variable Automatic Computer)，即“电子离散变量自动计算机”。冯·诺依曼思想以及 EDVAC 的出现在理论与实践上为现代电子计算机的体系结构与工作原理奠定了基础。冯·诺依曼本人因此被誉为“现代计算机之父”。

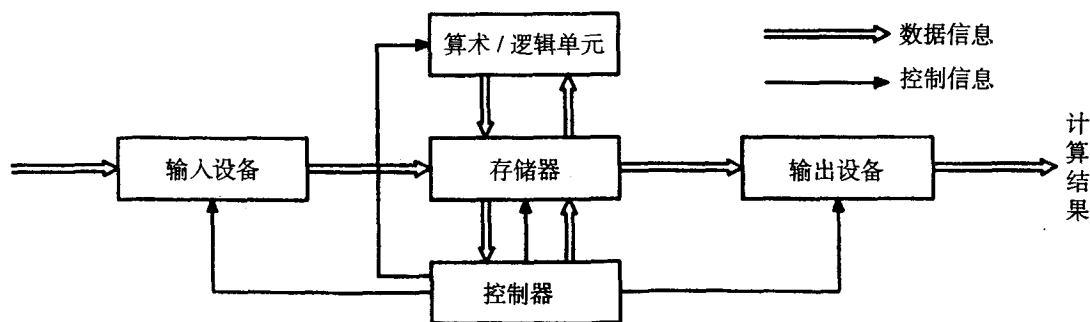


图 1.1 冯·诺依曼计算机硬件结构

从第一台电子计算机 ENIAC 诞生起，经过半个多世纪的发展，电子计算机的生产制造技术已经有了巨大进步，并且仍然继续得到快速发展。

计算机硬件更新的总体趋势是体积缩小、重量减轻、速度提高、成本降低和可靠性增加。据统计,每隔 5~7 年,电子计算机的运算速度提高 10 倍,可靠性提高 10 倍,成本降低到原来的 1/10。这种发展速度是任何其他行业所无法比拟的。到今天,计算机的应用由原来比较单一的科学计算深入到了工业生产、教育、军事、通信、管理和娱乐等领域,并且极大地改变了人类的社会生活和经济生活。

电子计算机需要在程序的指挥下才能正常工作,没有程序的计算机称为“裸机”,“裸机”是毫无用处的“废铁”,它根本无法为人们的生产和生活提供服务。因此,只有为计算机配置程序才能正确使用计算机。

要为计算机配置解决某种问题、实现某项功能、完成一定任务的程序就必须人工编制程序。编制程序的方法是使用计算机语言根据问题求解或者操作的步骤来描述(编写)程序。

计算机语言也称为程序设计语言(简称程序语言或编程语言),它是专门用来描述计算机程序的符合一定规范的文字符号系统。计算机程序中,描述(编写)程序的规范称为语法(syntax);而求解问题或者完成操作的步骤称为算法(algorithm)。算法是程序表达的语义,是程序的“灵魂”。算法必须用计算机语言来描述才能成为程序,所以,计算机语言也可以称为算法设计语言(简称为算法语言)。

计算机语言与人类的自然语言一样,具有多样性。在自然语言中,用不同的语言可以表达同样的意思,而程序设计中,同样的算法也可以用不同的编程语言来实现。在自然语言中,用同一种语言表达相同语义时,也可能因人而异地出现不同的表达方式。同样地,当用同一种计算机语言来描述相同的算法时,不同编程者描述出的程序也可能存在差异。

人们编程所用的语言,大体上经历了 4 个发展阶段,即机器语言、汇编语言、高级语言和集成开发工具。人们编制程序的方法,也由最初的依靠编程者的个人创意,发展到遵循一定规范和编程原则的结构化程序设计方法;由早期单一的面向过程的编程模式,发展到面向对象的编程模式;程序的生产、制作由早期的个体开发模式,转向适应大型程序生产的软件工程模式;程序正确性、有效性、可靠性以及效率的评价产生了量化指标以及系统化的控制方法。

1.1.1 机器语言

机器语言是一种二进制编码,它是计算机控制器能够直接接受的“语言”。

机器语言的基本组成单位称为机器指令或机器码。机器指令一般由操作码与操作数两部分组成。机器指令中的操作码用来表达操作的含义,即通知控制器该做什么;操作数用来描述操作所需的数据或数据的存储地址。机器语言程序就是由若干条机器指令组成的集合。

机器指令的二进制编码格式由硬件设计者在设计控制器时设计和规定。不同型号的计算机,如果它们的控制器不同,则相应的机器语言也不同。

对于特定型号的计算机,它所使用的全部机器指令构成的集合称为指令集。指令集中,指令格式的多样性、复杂性与指令的数量在一定程度上能够反映计算机硬件系统的性能。一般说来,指令集中指令格式的多样性越强、复杂程度越高、数量越多,说明硬件系统的性能指标越高。

现代计算机一般把控制器和算术/逻辑单元设计成一块芯片,称为中央处理单元,即 CPU (Central Processing Unit)。机器语言是 CPU 能够直接“理解”的惟一语言。任何用其他编程语言描述的程序,最终需要翻译成机器语言程序并存储于内存后,才能被 CPU 执行。

早期的一些计算机,只能接受机器语言程序,输入程序采用纸带穿孔的方法(如有孔表示 1,无孔表示 0),通过纸带阅读机直接输入二进制的机器语言程序和数据。

不同型号的计算机具有不同的机器语言,按某种机器语言编写的程序,只能用于这种特定型号的计算机,换用另一种型号的计算机,就得改用相应的机器语言重新编写程序。

由于机器语言是一种二进制编码,用机器语言写出的程序不仅可读性很差,而且检查和修改十分困难。此外,让编程者学习、记忆和掌握二进制的机器指令也不是一件容易的事。因此,用机器语言直接描述程序的方法很快被淘汰,取而代之的是汇编语言和高级语言。

1.1.2 汇编语言

汇编语言也称为符号语言,它采用英语单词或英语单词缩写、标点符号以及数字符号来描述机器指令。比如 IBM-PC 机采用的 8086/8088CPU 指令集中,用汇编语言描述的一条加法指令 ADD AX, 1 代表的机器指令是寄存器 AX 加 1,然后把运算结果存回寄存器 AX 中。该指令中,ADD 这个英文单词形象地描述了指令的功能,它代表指令的操作码,一般称为指令助记符,AX 和 1 代表指令的操作数。

显然,用这种接近自然语言的形式来描述机器指令,使得指令的学习和记忆更加方便,程序的可读性和易修改性也得到很大提高。

用汇编语言编写的程序是不能被计算机的 CPU 直接识别和接受的,它必须经过一个翻译过程,使之成为机器码程序后,才能被执行。实现这个翻译过程的程序称为汇编程序 (Assembling Program, 简称为 Assembler)。汇编程序是一种系统软件。

在 IBM-PC 系列微机中,编写汇编语言程序的过程如图 1.2 所示。

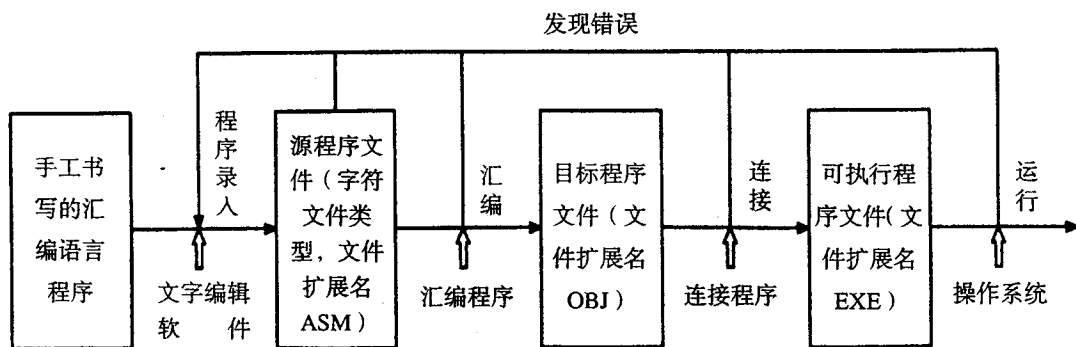


图 1.2 汇编语言的编程过程

汇编语言的编程过程首先是手工书写汇编语言程序,然后通过文字编辑软件录入程序,将程序存于磁盘,形成源程序 (Source Program) 文件。编程者接下来需要执行汇编程序对