

计算机与信息技术专业应用教材

# 程序设计基础

(基于C语言)

孙承爱 赵卫东 主编



清华大学出版社

TP312/2791

2008

# 程序设计基础

(基于 C 语言)

主编: 孙承爱 赵卫东

编委: 崔焕庆 孙红梅

鲁法明 吴振寰



清华大学出版社

北京

## 内 容 简 介

本书以 C 语言为操作语言, 全面介绍程序设计的入门知识, 是进一步学习其他计算机课程的基础。

全书共 12 章。第 1 章简介了计算机系统功能、基本组成和工作原理; 第 2 章介绍了使用 C 语言开发程序的过程、方法、工具和步骤; 第 3~5 章讲述了 C 语言基本数据类型、常用标准库函数, 以及顺序、选择、循环等结构; 第 6~7 章的内容是关于数组及函数的声明、定义和调用方法; 第 8 章介绍了宏、文件包含和条件编译等命令; 第 9~10 章讲述了地址、指针和结构体; 第 11 章是关于流及打开、关闭和读写文件的函数; 第 12 章完整地实现了一个学生成绩管理系统的设计。本书在附录中还配有 16 个实验, 供读者参考使用。

本书注重知识内容、综合练习和课程设计的有机统一, 循序渐进, 通俗易懂, 可作为高等学校 C 语言程序设计的教材, 特别适合作为应用型本科、高职院校的计算机及相关专业的 C 语言程序设计的教材, 同时也可作为编程人员和 C 语言自学者的参考用书。

在网站 (<http://www.khp.com.cn>) 上本书配有电子教案、8 套考试模拟题及其参考答案、课后习题参考答案、实验指导书及所列实验的实验参考步骤等附加内容供读者参考使用。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 翻印必究。侵权举报电话: 010-62782989 13501256678 13801310933

### 图书在版编目 (CIP) 数据

程序设计基础: 基于 C 语言/孙承爱, 赵卫东编著.  
—北京: 清华大学出版社, 2008.1

ISBN 978-7-302-16628-3

I. 程… II. ①孙… ②赵… III. C 语言—程序设计  
IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 195691 号

责任编辑: 刘秀青

责任校对: 刘雪莲

责任印制: 科海

出版发行: 清华大学出版社

<http://www.tup.com.cn>

[c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

社总机: 010-62770175

投稿咨询: 010-62772015

地 址: 北京清华大学学研大厦 A 座

邮 编: 100084

邮购热线: 010-62786544

客户服务: 010-62776969

印装者: 北京市鑫山源印刷有限责任公司

经 销: 全国新华书店

开 本: 185×260 印 张: 26

字 数: 633 千字

版 次: 2008 年 2 月第 1 版

印 次: 2008 年 2 月第 1 次印刷

印 数: 1~4 000

定 价: 39.80 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系  
调换。联系电话: (010) 82896445 转 8501 产品编号: 028016-01

# 前 言

学习程序设计的目的就是在掌握程序设计技术、方法和工具的基础上,采用计算机语言高效地编写出求解某一问题的高质量的计算机程序。而《程序设计基础》是学习程序设计的入门课程,是进一步学习《面向对象程序设计》、《数据结构》、《算法设计与分析》等课程的基础。C语言是目前最流行和使用最广泛的计算机语言之一,具有表达能力强、概念和功能丰富、目标程序质量高、可移植性好、使用灵活方便等特点,既具有高级语言的优点,又具有低级语言的某些特点,因此,特别适合于编写系统软件和嵌入式软件。C语言的上述特点使得它不仅在国内众多软件企业中得到广泛认可和应用,而且,我国绝大部分高等院校都把C语言作为计算机和非计算机专业的第一门程序设计语言课程。全国计算机等级考试、全国计算机应用技术证书考试也都将C语言列入了考试范围。

本书在详细阐述程序设计基本概念、原理和方法的基础上,通过大量经典实例重点讲解了C语言的概念、规则和使用方法,使程序设计语言的初学者能够在建立正确程序设计观念的前提下,掌握利用C语言进行结构化程序设计的技术和方法。

本书共分12章。

第1章简介了计算机系统的功能、基本组成、工作原理以及程序设计语言发展的历史;讲述了各种类型的数据及其在计算机内部表示的原理和方法;阐述了人类进行问题求解的过程、算法概念和表示方法,以及程序质量优劣的标准和各种质量要素的含义。

第2章简介了C语言发展的历史和C语言程序的组成及特点,详细讲述了使用Visual C++ 6.0集成开发环境开发C语言程序的过程,并重点介绍了编写和调试控制台程序的方法、工具和步骤。

第3章详细讲述了C语言标识符的概念以及基本数据类型和它们之间的相互转换,突出了常量与变量的概念、各种运算符及表达式的使用方法。

第4章讲述了C语言标准库函数的概念及其分类,详细讲解了常用的输入输出函数、字符串操作函数和数学运算函数,并对图形操作函数进行了简单介绍。

第5章讲述了C程序的组成及C语句的分类和编程风格,详细讲解了运用C语言进行顺序、选择、循环等结构进行编程的方法。

第6章讲述了数组的概念及其在内存中的存储结构,详细讲解了一维数组、二维数组的

定义、初始化和引用方法，字符数组的定义、初始化和使用方法；突出了字符串与字符数组的区别和联系，以及使用数组及字符串编程解决实际问题的方法。

第7章详细讲述了函数的声明、定义和调用方法；讲解了嵌套函数调用和递归函数调用的方法，突出了使用数组作为函数参数；简述了变量的作用域、可见性和生存期。

第8章讲述了宏的定义方法及应用文件包含命令，简述了条件编译命令。

第9章详细介绍了地址和指针的概念以及对各种类型的指针变量进行定义和引用的方法，简介了指针变量运算的含义；讲述了数组在内存中的存储结构以及运用指针操作数组和字符串的方法与技巧；简介了指针数组和指向指针的指针以及运用指针作为函数参数和返回值，同时简介了指向函数的指针的概念及其定义与应用方法。

第10章详细讲述了结构体的定义和使用方法以及链表的概念和相关操作，简介了共用体和枚举型数据。

第11章讲述了C语言标准库对输入输出的抽象——流的概念，详细讲解了以二进制方式访问文件和以文本方式访问文件的区别，突出了打开、关闭和读写文件的C语言标准库函数；简介了文件错误检查等其他与文件相关的C语言标准库函数。

第12章从一个学生成绩管理系统的实际需求出发，按照软件开发的思路，沿着需求分析→总体设计→详细设计→代码实现的过程，完整地进行了实现。该应用案例综合运用了本书所学的所有知识，尤其突出了结构体和文件操作以及各种常用算法的运用。整个第12章的流程图均采用传统流程图画法，目的在于使读者在掌握N-S图的同时，还能掌握传统流程图的画法；第12章还以Turbo C 2.0作为开发环境，利用Turbo C所提供的图形函数绘制图形，目的是为给还在使用Turbo C的用户提供更多的帮助。

本书附录配有16个实验，可供读者参考使用，相关实验步骤在网站(<http://www.khp.com.cn>)上下载。

本书作为程序设计的入门课程，重视对程序设计和C语言基本概念、原理和规则的讲解，力求给读者打下一个扎实的基础，培养读者良好的编程风格和工程纪律，提高读者进一步学习新程序设计语言的能力。

本书采用教案式编写方式，通过每一章的“引言”给出各章要解决的问题和主要的学习内容，明确学习目标；对每一个知识点，首先阐述相关的概念，然后通过实例加以准确清晰的说明，并给出有针对性的思考题，引导读者逐步加深对知识点的理解；C语言所有语法规则都采取“格式—功能—举例—说明”的方式进行详细介绍，对于学生容易出错的地方给出应注意的事项。书中每一道程序设计例题都给出算法描述和相关知识点的分析，并注重一题多解，提高读者分析问题和设计算法的能力；全书自始至终通过“学生成绩管理系统”的不断完善，详细阐述各个知识点的应用方法，提高学生运用所学知识解决实际问题的能力。

C语言涉及的概念多、规则复杂、容易出错，初学者往往感觉困难。本书采用循序渐进、深入浅出、通俗易懂的讲解方法，本着理论与实际相结合的原则，注重知识内容、综合练习和课程设计的有机统一，重在培养学生的实际操作能力和程序设计能力，最后达到掌握C语言程序设计的目的。

本书可作为高等学校各专业C语言程序设计的教材，特别适合作为应用型本科、高职院校的计算机及相关专业的学生学习C语言程序设计的教材，同时也可作为编程人员和C语言自学者的参考用书。

全书由郑永果教授审校。第1章由赵卫东编写；第2章、第5章、第7章、第8章由崔焕庆编写；第3章、第12章由孙承爱编写；第4章由孙红梅编写；第6章、第10章由鲁法明编写；第11章由吴振寰编写；第9章由大家共同编写。在编写的过程中，范明芳老师和学生王程、侯明雷也为教材的编写做了许多工作，在此一并表示感谢！最后还要感谢为本书付出心血的编辑、阅稿人员等各位朋友！

在网站 (<http://www.khp.com.cn>) 上本书配有电子教案，8套考试模拟题及其参考答案，课后习题参考答案，实验指导书及所列实验的实验参考步骤等附加内容供读者参考使用。

由于作者水平有限，书中难免出现遗漏和不足之处，恳请社会业界同仁及读者朋友提出宝贵意见和真诚的批评，E-mail请寄：[feedback@khp.com.cn](mailto:feedback@khp.com.cn)或[sun910213@163.com](mailto:sun910213@163.com)

编者

2008年1月

# 目 录

<b>第1章 绪论</b> .....	1	2.2.1 C语言程序组成	48
1.1 计算机系统概述	1	2.2.2 C语言程序的特点	50
1.1.1 计算机系统功能	2	2.3 C语言程序上机指导	51
1.1.2 计算机系统组成	2	2.3.1 执行C程序的步骤	51
1.1.3 计算机工作原理	8	2.3.2 Visual C++ 6.0 集成开发环境	52
1.2 数据及其计算机内部表示	9	2.4 C程序的调试	55
1.2.1 与数据相关的概念和术语	9	2.4.1 程序调试策略与技术	55
1.2.2 二进制表示法	10	2.4.2 VC6下的程序调试方法	57
1.2.3 文本数据表示	11	本章小结	63
1.2.4 数值数据表示	13	习题2	64
1.2.5 音频数据表示法	16	<b>第3章 基本数据类型与数据运算</b>	<b>65</b>
1.2.6 图形和图像数据表示法	16	3.1 基本标识符	65
1.2.7 视频数据表示法	18	3.1.1 保留关键字	65
1.2.8 数据及其表示小结	18	3.1.2 预定义标识符	66
1.3 问题求解与算法	18	3.1.3 用户自定义标识符	66
1.3.1 问题求解	19	3.2 数据类型	66
1.3.2 算法及其特点	21	3.3 常量与变量	68
1.3.3 算法优劣标准	21	3.3.1 常量	68
1.3.4 算法描述	22	3.3.2 变量	72
1.4 程序设计与程序设计语言	32	3.4 基本数据类型的转换	75
1.4.1 程序质量	33	3.4.1 自动类型转换	75
1.4.2 程序设计语言的发展史	35	3.4.2 强制类型转换	75
1.4.3 结构化程序设计方法	39	3.5 运算符和表达式	76
1.4.4 程序设计必备知识	42	3.5.1 运算符和表达式概述	76
1.5 计算机问题求解的过程	43	3.5.2 算术运算符与算术表达式	77
1.5.1 算法开发	43	3.5.3 赋值运算符与赋值表达式	79
1.5.2 算法实现	43	3.5.4 关系运算符与关系表达式	81
本章小结	44	3.5.5 逻辑运算符与逻辑表达式	82
习题1	45	3.5.6 条件运算符与条件表达式	83
<b>第2章 C语言概述</b>	<b>47</b>	3.5.7 逗号运算符与逗号表达式	83
2.1 C语言发展史	47	3.5.8 位运算符与位运算表达式	84
2.2 C语言程序组成及特点	48	3.5.9 取长度运算符	88
		3.5.10 运算符的优先级和结合性	89

3.6 应用举例 .....	90	5.3.1 if语句 .....	123
本章小结 .....	92	5.3.2 switch语句 .....	127
习题3 .....	93	5.3.3 选择程序举例 .....	129
<b>第4章 常用标准库函数 .....</b>	<b>96</b>	5.4 循环结构 .....	132
4.1 输入输出函数 .....	96	5.4.1 while语句 .....	132
4.1.1 字符输入输出函数 .....	97	5.4.2 do-while语句 .....	132
4.1.2 格式输入输出函数 .....	98	5.4.3 for语句 .....	133
4.2 字符串操作函数 .....	103	5.4.4 跳转语句 .....	136
4.2.1 字符串输出函数puts() .....	104	5.4.5 循环的嵌套 .....	138
4.2.2 字符串输入函数gets() .....	104	5.4.6 循环程序举例 .....	140
4.2.3 字符串连接函数strcat() .....	104	5.5 应用举例 .....	142
4.2.4 字符串复制函数strcpy() .....	105	本章小结 .....	145
4.2.5 字符串比较函数strcmp() .....	105	习题5 .....	146
4.2.6 求字符串长度函数strlen() .....	106	<b>第6章 数组 .....</b>	<b>149</b>
4.2.7 strlwr函数 .....	107	6.1 数组的定义和存储结构 .....	149
4.2.8strupr函数 .....	107	6.1.1 数组的定义 .....	149
4.3 数学运算函数 .....	108	6.1.2 数组的存储结构 .....	150
4.3.1 求绝对值函数 .....	108	6.2 一维数组 .....	151
4.3.2 指数和对数函数 .....	108	6.2.1 一维数组的定义 .....	151
4.3.3 三角函数 .....	109	6.2.2 一维数组元素的引用 .....	151
4.3.4 其他函数 .....	110	6.2.3 一维数组的初始化 .....	152
4.4 图形操作函数 .....	111	6.2.4 一维数组程序举例 .....	152
4.4.1 有关图形模式和坐标位置的 函数 .....	111	6.3 二维数组 .....	157
4.4.2 屏幕颜色的设置和清屏函数 .....	111	6.3.1 二维数组的定义 .....	157
4.4.3 基本图形函数 .....	113	6.3.2 二维数组元素的引用 .....	157
4.4.4 封闭图形填充函数 .....	115	6.3.3 二维数组的初始化 .....	157
4.5 综合应用举例 .....	116	6.3.4 二维数组程序举例 .....	158
本章小结 .....	117	6.4 字符串与字符数组 .....	159
习题4 .....	117	6.4.1 字符串 .....	159
<b>第5章 C程序结构及控制语句 .....</b>	<b>118</b>	6.4.2 字符数组 .....	160
5.1 C程序语句及编程风格 .....	118	6.4.3 字符串与字符数组程序举例 .....	163
5.1.1 C程序语句 .....	118	6.5 应用举例 .....	165
5.1.2 编程风格 .....	120	本章小结 .....	167
5.2 顺序结构 .....	121	习题6 .....	168
5.3 选择结构 .....	123	<b>第7章 函数 .....</b>	<b>170</b>
		7.1 引言 .....	170



7.2 函数的定义 .....	171	9.2.2 指针变量的定义及引用 .....	224
7.3 函数的调用 .....	173	9.2.3 指针变量作函数参数 .....	228
7.3.1 函数调用 .....	173	9.2.4 指向指针的指针 .....	231
7.3.2 函数的参数 .....	174	9.2.5 指针的运算 .....	232
7.3.3 函数的返回值 .....	175	9.2.6 void指针类型 .....	233
7.3.4 函数的声明 .....	176	9.3 指针与数组 .....	234
7.4 嵌套与递归 .....	179	9.3.1 数组的指针 .....	234
7.4.1 函数的嵌套调用 .....	179	9.3.2 指向数组元素的指针 .....	235
7.4.2 函数的递归调用 .....	181	9.3.3 指向数组的指针 .....	237
7.5 数组作为函数参数 .....	184	9.3.4 数组指针作参数 .....	239
7.5.1 数组元素作为函数参数 .....	184	9.4 指针与字符串 .....	243
7.5.2 数组名作为函数参数 .....	185	9.4.1 字符串的字符指针表示形式 .....	243
7.5.3 多维数组名作为函数参数 .....	187	9.4.2 利用字符指针访问字符串 .....	244
7.6 变量的作用域与生存期 .....	188	9.4.3 字符数组与字符指针的比较 .....	246
7.6.1 变量的作用域 .....	189	9.4.4 字符指针作函数参数 .....	248
7.6.2 变量的生存期 .....	191	9.5 指针数组 .....	250
7.7 内部函数与外部函数 .....	197	9.5.1 一维指针数组的定义及使用 .....	250
7.7.1 内部函数 .....	197	9.5.2 指针数组作main函数的形参 .....	251
7.7.2 外部函数 .....	197	9.6 指针与函数 .....	252
7.8 应用举例 .....	198	9.6.1 返回指针值的函数 .....	252
本章小结 .....	204	9.6.2 指向函数的指针 .....	255
习题7 .....	206	9.6.3 函数指针作为函数的参数 .....	257
<b>第8章 预处理命令 .....</b>	<b>211</b>	9.7 应用举例 .....	260
8.1 宏定义 .....	211	本章小结 .....	264
8.1.1 无参数宏 .....	211	习题9 .....	265
8.1.2 带参数的宏 .....	214	<b>第10章 结构体与共用体 .....</b>	<b>269</b>
8.2 文件包含 .....	216	10.1 结构体 .....	269
8.3 条件编译 .....	217	10.1.1 结构体类型的定义 .....	269
8.3.1 条件编译的格式 .....	217	10.1.2 结构体变量的定义、引用和 初始化 .....	270
8.3.2 使用条件编译的优点 .....	219	10.1.3 结构体程序举例 .....	272
本章小结 .....	220	10.2 链表 .....	277
习题8 .....	221	10.2.1 链表概述 .....	277
<b>第9章 指针 .....</b>	<b>222</b>	10.2.2 静态链表 .....	278
9.1 引言 .....	222	10.2.3 动态链表 .....	279
9.2 指针与指针变量 .....	223	10.3 共用体 .....	285
9.2.1 指针的概念 .....	223	10.3.1 共用体的概念 .....	285

10.3.2 共用体的定义 .....	286	11.4 应用举例 .....	319
10.3.3 共用体变量的引用 .....	287	本章小结 .....	320
10.3.4 共用体应用举例 .....	287	习题11 .....	320
10.4 枚举类型 .....	288	<b>第12章 综合应用案例 .....</b>	<b>322</b>
10.4.1 枚举的概念 .....	288	12.1 需求陈述 .....	322
10.4.2 枚举的定义 .....	288	12.2 需求分析 .....	322
10.4.3 枚举变量 .....	289	12.2.1 功能需求 .....	322
10.4.4 枚举应用举例 .....	290	12.2.2 数据需求 .....	323
10.5 应用举例 .....	291	12.2.3 技术约束 .....	323
本章小结 .....	295	12.3 总体设计 .....	324
习题10 .....	296	12.3.1 系统总体结构 .....	324
<b>第11章 文件 .....</b>	<b>298</b>	12.3.2 全局数据结构 .....	325
11.1 引言 .....	298	12.3.3 界面设计 .....	325
11.2 流和文件指针 .....	299	12.4 详细设计 .....	336
11.2.1 流 .....	299	12.4.1 管理员部分 .....	337
11.2.2 文件指针 .....	301	12.4.2 学生部分 .....	355
11.3 文件的操作 .....	302	12.5 完整代码 .....	358
11.3.1 fopen函数和fclose函数 .....	302	习题12 .....	392
11.3.2 fgetc函数和fputc函数 .....	305	<b>附录A 常用字符与ASCII代码对照表 ..</b>	<b>393</b>
11.3.3 fgets函数和fputs函数 .....	309	<b>附录B C语言ANSI/ISO标准库函数 ..</b>	<b>394</b>
11.3.4 fread函数和fwrite函数 .....	311	<b>附录C TC环境下的图形函数 .....</b>	<b>399</b>
11.3.5 fprintf函数和fscanf函数 .....	313	<b>附录D 实验 .....</b>	<b>400</b>
11.3.6 其他读写函数 .....	314	<b>参考文献 .....</b>	<b>406</b>
11.3.7 文件的定位函数 .....	315		
11.3.8 出错检测函数 .....	316		
11.3.9 标准库中其他文件操作函数 ..	317		

# 第 1 章

## 绪 论

### CHAPTER 01

**学习目标** 在了解计算机系统功能和基本组成的基础上,正确理解计算机系统的工作原理;了解各种类型的数据及其在计算机内部表示原理和方法;了解人类进行问题求解的目的和过程;了解程序设计语言发展的历史。理解程序质量优劣的标准和各种质量要素的含义;重点掌握主存储器的结构,字符的ASCII码、整数的补码和实数的表示方法,算法的概念、特点和各种描述方法,计算机程序、程序设计和程序设计语言的概念,结构化程序设计方法思想和原则,以及利用计算机进行问题求解的过程。

在计算机硬件相对固定不变的前提下,要控制计算机(Computer)给人们提供各种各样的服务,就必须为其编写不同的计算机程序(Program)。而学习程序设计(Programming)的目的就是在掌握程序设计方法、技术和工具的基础上,采用计算机语言高效地编写出求解某一问题的高质量的计算机程序。为了达到上述目标,需要经过系统化的学习和大量的上机实践。《程序设计基础》是学习程序设计的入门课程,是进一步学习《面向对象程序设计》、《数据结构》、《算法设计与分析》等课程的基础。

计算机程序是计算机系统的组成部件之一,与计算机系统的其他部件之间有着紧密的关系,因此,学习程序设计就必须首先掌握计算机系统的功能、组成和工作原理,掌握程序的概念及其在计算机系统中所发挥的重要作用。数据是程序处理的对象,算法是程序对数据处理步骤的过程描述,是程序的灵魂,因此,数据、数据在计算机内部的表示,以及算法的概念、特征及其描述方法是本章重点讲述的内容之一。要高效地编写出高质量的程序,就必须理解程序质量优劣的标准、科学的程序设计方法和计算机语言的有关知识,因此,本章在阐述程序质量要素概念的基础上,讨论了提高程序质量的技术措施,总结了程序设计语言发展历史及其各种语言的主要特点,通过实例重点讲述了结构化程序设计思想、概念和原理。本章最后总结了利用计算机进行问题求解的详细过程。

### 1.1 计算机系统概述

当今世界,计算机作为被广泛应用的能完成各种计算任务的通用机器,在各行各业都发挥了巨大的作用,已成为社会和经济发展的“加速器”和“催化剂”。没有计算机的支持,很多行业的具体工作已不能正常开展。

在很多人眼里,计算机是一个“神秘”的机器,具有很强的通用性,似乎无所不能。其实,在计算机硬件相对固定不变的前提下,计算机的通用性主要表现在通过运行不同的程序来完成不同的计算任务,由此可见程序设计的重要性。

本节采用由表及里循序渐进的论述方式,重点选取与程序设计有关的内容,从计算机系统功能、组成和工作原理3个方面来剖析计算机系统,使读者对计算机系统有一个宏观的认识和了解,为更深入地学习程序设计技术奠定良好的基础。

### 1.1.1 计算机系统功能

如果把计算机看作一个黑盒子,即不关心其内部结构,只从其外部功能特征出发,我们可以把计算机系统抽象成一个如图1.1所示的通用的数据处理器。其主要具有数据输入、处理、存储和输出4大基本功能。在特定程序的控制下,计算机接收若干输入(Input)数据,并按照程序所规定的步骤、规则和方法对数据进行处理以产生所期望的输出(Output)数据;借助存储设备,计算机可以把输入输出数据以及中间计算结果永久保存。

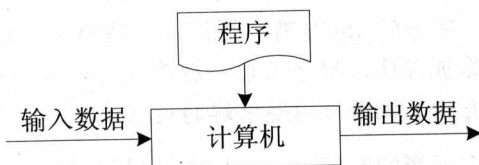


图 1.1 计算机功能视图

图1.1中,程序是用来控制计算机完成数据输入、处理、输出和存储功能的指令集合(Instruction Set),而指令是用来执行某种具体操作的命令,如各种算术运算和逻辑运算指令。由于计算机系统中一条指令只能完成非常有限的功能,因此,为了完成某项计算任务,常常需要有一系列指令的组合,所以程序是指令的集合。

在早期的计算机中,指令是通过配线或一系列开关的物理变换来实现的,显然,采用这种方式来编写和执行大量不同的指令是非常麻烦和困难的。而采用各种计算机语言,即各种符号编写不同的程序来控制计算机完成不同的计算任务则容易得多。计算机可以在硬件不变的基础上,理解不同的程序并按程序的规定自动完成相应的计算任务。因此,从这一角度,我们可以把计算机程序定义为:“用计算机语言所编写的一系列指令的集合”。

计算机的通用性就表现在可以编写不同的程序来控制计算机完成不同的计算任务。就像是一个交响乐团可以按照不同的乐谱演奏出不同的音乐,这里乐谱就是控制乐团成员演奏的一系列指令集合(程序),它规定了演奏的过程和方法。

### 1.1.2 计算机系统组成

按照“系统论”的观点,一个复杂的系统是由若干既相互独立又相互联系、相互作用的部件所组成的一个有机整体。计算机系统作为一个复杂的人工系统,从总体上看,是由硬件系统和软件系统两大部分组成的,这两部分相互联系、相互作用,共同完成计算机系统的功能。

## 1. 计算机硬件系统

计算机硬件是指构成计算机系统的各种物理装置，是看得见、摸得着的一些实实在在的有形实体。例如，键盘、鼠标和显示器等。为了完成计算机系统的4大基本功能，如图1.2所示，任何计算机系统的硬件都必须由运算器、控制器、存储器、输入设备和输出设备5大基本部件构成，少掉其中任何一个部件就不能叫计算机了。

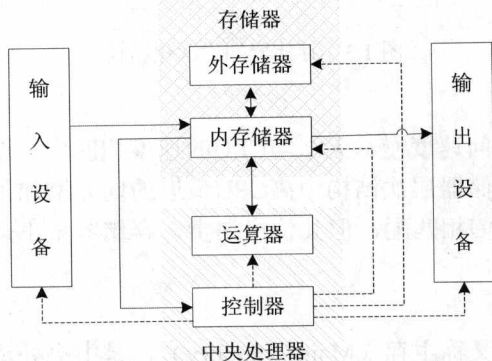


图 1.2 计算机硬件的基本组成

图中实线箭头代表数据或指令，在机器内用二进制数来表示。虚线箭头代表控制信号，在机器内呈现高低电平形式，起控制作用。这是两种不同类型的信息，计算机的工作正是通过这两种不同性质的信息流动完成的。下面围绕此图说明硬件各组成部件的作用以及它们是如何配合工作的。

### (1) 输入/输出设备 (Input/Output device)

输入设备是用来输入程序和数据的物理部件。常见的输入设备包括键盘 (Keyboard)、鼠标 (Mouse)、麦克风 (Microphone)、扫描仪 (Scanner)、数字化仪 (Digitizer)、数码相机 (Digital Camera)、触摸屏 (Touch Screen) 等。各种输入设备能够接收包括数字、文字、图像、音频和视频等多种形式的数 据，并将其转换成计算机内部统一的表示方法，然后存储到存储器中。我们将在 1.2 节详细阐述各种类型数据在计算机内部的表示方法。

输出设备是用来输出数据的物理部件。常用的输出设备包括显示器 (Monitor)、打印机 (Printer)、绘图仪 (Plotter)、音箱 (Sound Box) 等。各种输出设备将计算机内部存储的数据转换成人们所能接受的各种形式输出。

由于输入输出设备都是些机电、磁性或光学设备，与中央处理器和内存相比，其操作的速度要慢得多，为了避免速度不匹配所造成的资源浪费，I/O 设备必须通过被称为 I/O 控制器或接口的器件来处理这种差异。例如，显示器必须通过显示控制器（俗称显卡）与计算机主机相连。

### (2) 存储器 (Memory)

存储器是计算机中具有记忆能力的物理部件，用来存放程序或数据。程序和数据是两种不同的信息，应放在不同的地方，两者不可混淆。计算机存储器的大小以字节 (Byte) 来计量，其中，1B (Byte) = 8b (Bit)，1KB (Kilobyte) = 1024B、1MB (Megabyte) = 1024KB，

1GB (Gigabyte) = 1024MB, 1TB (Terabyte) = 1024GB。

为了提高中央处理器 (Central Processing Unit, CPU) 的利用率, 在计算机系统中, 存储器常采用层次化结构。存储器通常可分为3大类, 即高速缓存 (Cache)、内存储器 (Main Memory) 和外存储器 (External Memory), 如图1.3所示。

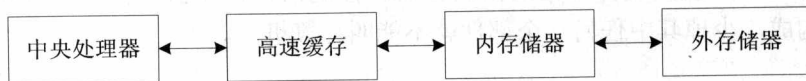


图 1.3 存储器的层次化结构

### ① 高速缓存

由于内存和外存的访问速度慢, 无法与CPU的速度相匹配, 造成CPU的利用率低下。为了解决这一问题, 在存储器层次结构中离CPU最近的地方增加了Cache, Cache的最大特点是其速度完全可以与CPU相匹配, 但其价格昂贵, 存储容量小, 因此, 通常将使用最频繁的数据存入Cache。

### ② 内存储器

内存储器简称内存, 又称主存 (Main Memory), 是中央处理器能根据地址线直接寻址的存储空间, 由半导体器件制成。其特点是比外存的存取速度快, 比Cache的容量大。所有要执行的程序都必须装入内存。

内存按其功能和存储信息的原理又可分成两大类, 即随机存储器 (Random Access Memory, RAM) 和只读存储器 (Read Only Memory, ROM)。RAM是一种在计算机正常工作时可读/写的存储器, 掉电会丢失信息, 因此, 用户在操作电脑过程中应养成随时存盘的习惯, 以防断电丢失数据。ROM与RAM的不同之处, 是ROM在计算机正常工作时只能从中读出信息, 利用这一特点常将操作系统基本输入输出系统 (Basic Input Output System, BIOS) 程序固化其中, 机器加电后立刻执行其中的程序。而用来存储用户程序和数据的内存是由RAM构成的。

计算机系统把内存看作是由若干个连续的存储单元 (Storage Location) 组成的, 每个存储单元的大小为1B。为了能唯一标识每个存储单元, 在计算机系统中给每个存储单元指定一个唯一的编号, 该编号被称为存储单元的地址 (Address), 计算机在读写内存时就是按照存储单元的地址进行的。假定地址总线为20b, 则可寻址的地址空间为00000ff~0xfffffH, 最大可寻址的存储单元个数为 $2^{20}$ 。其中, 每个存储单元都对应一个唯一的地址和实际存储的数据, 该数据称为存储单元的内容。如图1.4所示, 地址为00002H的存储单元的内容为11111001B, 记为[00002H]=11111001B。

### ③ 外存储器

外存储器简称外存 (External Memory), 它作为一种辅助存储设备, 主要用来存放一些暂时不用而又需常期保存的程序或数据。当需要执行外存中的程序或处理外存中的数据时, 必须通过输入/输出指令将其调入内存中, 才能被CPU处理, 所以外存实际上属于输入/输出设备。外存的最大特点是存取速度相对较慢, 但容量可以做得很大, 价格比较便宜。例如, 硬盘容量可达几十或几百GB。外存可以长期保存大量程序或数据, 是计算机中必不可少的重要设备。

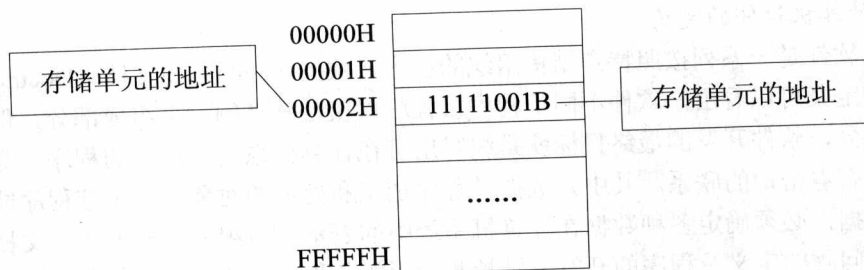


图 1.4 内存的结构

外存通常采用文件 (File) 形式来组织和存储数据。所谓文件, 就是可按名字 (称为文件名) 访问的相关数据的集合。当我们要将程序或数据存储到外存时, 首先必须按照操作系统 (Operating System, OS) 的规定为所要存储的程序和数据指定一个合法的文件名; 同样, 当我们要读取外存的数据时, 必须先按文件名找到指定的文件, 然后再从该文件读取数据。

### (3) 中央处理器

CPU是计算机系统的核心部件, 其主要功能是解释并执行指令, 其主要逻辑部件包括运算器、控制器和寄存器。计算机的性能主要取决于CPU。

#### ① 运算器 (Arithmetic Logic Unit, ALU)

运算器又称算术逻辑部件, 是计算机用来进行算术运算和逻辑运算的部件。运算器按照预定指令的要求通过地址取出内存中特定的操作数执行运算, 并将运算结果写入内存。

#### ② 控制器 (Control Unit)

控制器是计算机的指挥系统, 计算机的工作就是在控制器控制下有有条不紊协调工作的。控制器通过地址访问存储器, 逐条取出选中单元的指令, 分析指令, 根据指令产生相应的控制信号并作用于其他各个部件, 控制其他部件完成指令要求的操作。上述过程周而复始, 保证了计算机能自动、连续的工作。

#### ③ 寄存器 (Register)

寄存器是CPU内部高速独立的存储单元, 主要用来存储数据、指令和程序的运行状态。与其他的存储设备相比, 寄存器速度最快, 但数量非常有限。

## 2. 计算机软件

随着计算机的日益普及, 用户对软件的需求量越来越大, 质量要求越来越高, 软件在计算机系统中的作用和地位越来越重要, 软件所涉及的应用领域越来越广泛, 这导致了软件的规模越来越大, 复杂性越来越高。与此同时, 尽管软件构建与项目管理的技术和工具不断进步, 但直到今天为止, 总体上仍然未从根本上改变软件生产效率低下和质量不可靠的问题, 软件开发失败的案例多得惊人。本小节首先给出了软件的定义, 然后按不同的标准对软件进行了分类, 以反映软件应用的深度和广度, 最后从软件本身的特性出发, 分析为什么软件较之其他物理产品更难以开发和维护。

### (1) 计算机软件的定义

计算机软件是一系列按照特定结构组织的程序、数据 (Data) 和文档 (Document) 的集合。从该定义可以看出, 软件不同于程序, 程序仅仅是软件的一个组成部分, 但它是软件的核心成分, 软件开发的最终目标就是编写出可在计算机系统上执行的程序。数据和文档都与程序有着密切的联系。其中, 数据是程序加工和处理的对象, 为了使程序能有效地处理各种数据, 必须确定各种数据在计算机系统上的表示、组织和存储方式。文档是关于程序所求解问题的定义及程序的开发、维护和使用的各种图文资料, 这些图文资料对于正确开发和使用程序具有重要的作用, 是软件不可或缺的重要成分。

### (2) 计算机软件的分类

软件应用领域广泛, 种类繁多, 因此难以给出一个统一的分类标准, 只能从不同的角度对软件进行不同的分类。

#### ① 按照软件的服务对象

按照软件的服务对象, 我们可以把软件分为定制软件 (Custom Software)、通用软件 (Generic Software) 和嵌入式软件 (Embedded Software) 3 大类。

定制软件的服务对象是特定的用户, 是为了满足特定用户的需要而开发的专用软件, 其成功与否主要取决于能否满足特定用户的个性化需求, 这种软件对其他客户的用处不大。例如, 某机构的网站和各种管理信息系统等。

通用软件也被称为商业成品 (Commercial Off-The-Shelf, COTS) 软件, 其服务对象是潜在的市场用户群体, 软件需求主要由市场调研确定, 必须要满足广大用户群体的共同需要, 例如, 字处理软件、电子表格、浏览器和操作系统等。尽管通用软件价格便宜且质量更可靠, 但它很难在所用应用领域满足所有用户的个性化需求, 所以不可能完全替代定制软件。

嵌入式软件必须固化在特定的硬件设备中, 例如洗衣机、微波炉和汽车等。尽管嵌入式软件最终的服务对象与通用软件相同, 但如果不同时更新硬件, 用户通常不能更新嵌入式软件, 而且其开发方法与通用软件存在明显的区别。

#### ② 按照软件的作用

按软件的作用, 我们可以将软件划分为系统软件 (System Software)、应用软件 (Application Software) 和介于这两者之间的中间件 (Middleware) 3 大类。

系统软件是指管理、监控、维护计算机正常工作和供用户操作使用计算机的软件。这类软件一般与具体应用无关, 其目的是为计算机的一般用户提供最基本的功能, 为程序员提供应用程序编程接口 (Application Programming Interface, API)。例如操作系统、各种语言处理程序、数据库管理系统 (Database Management System, DBMS) 等。

应用软件种类繁多, 是指为满足某特定领域中某种具体应用要求而开发的软件, 如财务报表软件、各种数据库应用软件等。值得注意的是, 系统软件和应用软件之间并无严格的界限, 随着计算机应用的普及, 应用软件也在向标准化、商业化方向发展, 并将其纳入软件库中。这些软件库既可看成是系统软件, 也可视为应用软件。

中间件是位于系统平台 (硬件和操作系统) 和应用之间的通用服务, 这些服务具有标准的程序接口和协议。针对不同的操作系统和硬件平台, 它们可以有符合接口和协议规范



的多种实现。中间件主要的作用是为处于上层的应用软件提供运行与开发环境，以帮助用户灵活、高效地开发和集成复杂的应用软件，IBM公司WebSphere和BEA公司的WebLogic应用服务器就属于中间件。

### (3) 计算机软件的特点

软件是软件工程师头脑思维的产物，与硬件相比具有显著不同的特点，主要表现在如下几个方面：

#### ① 表现形式不同

硬件在其整个生命周期都表现为有形的、可见的物理实体，而软件则是存在于软件工程师的脑袋里或纸面上，在可执行的程序开发出来之前，软件表现为一种无形的、不可见的逻辑实体。因此，其研发进度和质量难以控制和管理，这无疑给软件的开发和维护工作带来许多困难。

#### ② 生产方式不同

软件是人力密集型的智力活动的结果，且每个新软件的开发常常是从零做起，可复用的已有成果很少，因此软件生产率低下，成本主要是开发和维护的人力成本；而其生产仅是一个简单的复制过程，与开发和维护相比，生产的成本几乎为零。这显然与传统意义上的硬件生产有着本质的区别。

#### ③ 维护方式不同

众所周知，如果硬件出现了故障或需要升级，则可以通过更换某些部件的方法来解决。但如果软件出现了运行错误或需要功能及性能的升级，则维护方式要复杂得多。软件维护的困难并不在于修改任务本身，而主要体现在理解软件和确定软件修改的位置、内容和方法所花费的代价上。对于一个规模庞大、结构复杂的软件来说，要完成上述两个任务是非常困难的。

### 3. 计算机系统的层次结构

如图1.5所示，计算机系统是按层次结构组织的。各层之间的关系是：内层是外层的支撑环境，而外层可不必了解内层细节，只需根据约定调用内层提供的服务。

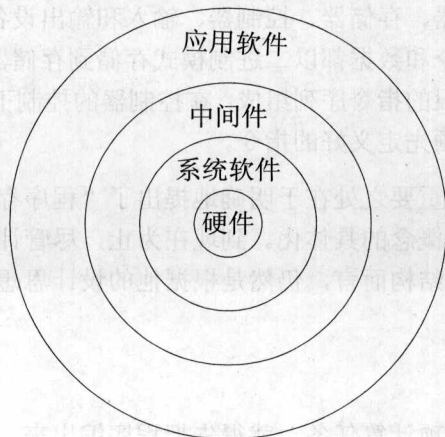


图 1.5 计算机系统层次结构