

程序设计基础

——基于C语言 (第2版)

孙承爱 赵卫东 主编

► 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

(第2版)

主编：孙承爱 赵卫东

编委：崔焕庆 孙红梅

鲁法明 吴振寰

金 兰 李园园

科学出版社

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宿舍卫生管理系统的设计。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,同时也可作为参加计算机等级考试的考生、编程人员和C语言自学者的参考用书。

本书为读者提供了程序源代码,读者可到网站 www.ncpress.com.cn 上免费下载。另外,本书有配套的学习辅导书——《程序设计基础(基于C语言)习题解答、上机指导、试题精选》(ISBN:978-7-300-10154-5),读者可参考使用。为方便教学,还特为本书的用书教师提供了教学资源包,供任课教师参考使用。

图书在版编目(CIP)数据

程序设计基础:基于C语言 / 孙承爱, 赵卫东主编.

—2版. —北京:科学出版社,2010.9

计算机与信息技术专业应用教材

ISBN 978-7-03-028797-7

I. ①程… II. ①孙…②赵… III. ①C语言-程序设计-高等学校-教材 IV. ①TP312

中国版本图书馆CIP数据核字(2010)第166897号

责任编辑:桂君莉 刘志燕 / 责任校对:杨慧芳

责任印刷:新世纪书局 / 封面设计:林 陶

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

中国科学出版集团新世纪书局策划

北京市艺辉印刷有限公司印刷

中国科学出版集团新世纪书局发行 各地新华书店经销

*

2010年10月第2版

开本:16开

2010年10月第一次印刷

印张:21

印数:1—3 000

字数:511 000

定价:36.00元

(如有印装质量问题,我社负责调换)

前言

本教材的第1版受到了广大读者的强烈好评，第2版是作者在第1版的基础上，本着与时俱进的原则，根据C语言的最新教学需要，结合教学、研究以及实际工程中积累的最新经验，对第1版的修订进行组织和编写的。

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。

本书共12章

第1章 讲解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 讲解了C的发展历史和程序的组成及特点，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，并重点描述了编写和调试控制台程序的工具、方法和步骤。

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。

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 详细讲述了地址和指针的概念，定义和引用指针变量的方法，讲解了指针变量的运算，数组在内存中的存储结构，使用指针操作数组和字符串的方法与技巧；讲解了指针数组和指向指针的指针，使用指针作为函数参数和返回值，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，介绍了共用体和枚举型数据。

第11章 讲述了C语言标准库对I/O的抽象——流的概念，详细讲解了以二进制和文本方式访问文件的区别，介绍了打开、关闭、读写文件、文件错误检查等文件操作类的标准库函数。

第12章 从一个学生宿舍卫生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→代码实现的软件开发流程，完整地展开系统的设计与实现。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。整个系统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 Visual C++ 6.0 作为开发环境，采用 Visual C++ 6.0 所提供的图形函数绘制图形。

另外，本书附录配有 16 个实验，可供读者参考使用。

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；
- 全书自始至终通过对“学生宿舍卫生管理系统”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。

本书的适用对象

本书可作为应用型本科、高职院校的计算机及相关专业、培训类学校的学生学习 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，本书与配套的辅导书——《程序设计基础（基于 C 语言）习题解答、上机指导、试题精选》尤其适合作为计算机等级考试 C 语言的考试用书。

本书的配套资源

本书有配套的学习辅导书——《程序设计基础（基于 C 语言）习题解答、上机指导、试题精选》(ISBN:978-7-300-10154-5)，读者可参考使用。

本书还为读者提供了程序源代码，读者可到 www.ncpress.com.cn 网站上免费下载。

另外，还为任课教师提供教学资源包【本书的电子课件，按章编排的习题库及其答案（约 600 多道题），8 套完整的考试模拟题及其参考答案，课后习题参考答案，实验指导书等内容】，供教师教学参考使用。

用书教师请发送 E-mail 至 bookservice@126.com 或致电 (010) 64865699 转 8033 免费索取本书的教学资源包。

编者寄语

全书由郑永果教授审校。第 1 章由赵卫东编写；第 2、5、7、8 章由崔焕庆编写；第 3、12 章由孙承爱编写；第 4 章由孙红梅编写；第 6、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、侯明雷为教材的编写做了许多工作，李园园和金兰老师在本书的改版升级中也做了大量的工作，在此一并表示感谢！

由于作者水平有限，书中难免出现遗漏和不足之处，恳请广大读者提出宝贵意见和真诚的批评，请发送 E-mail 至 bookservice@126.com 或 sun910213@163.com。

编 者
2010 年 9 月

目 录

第1章 绪论.....1

- 1.1 计算机系统概述.....1
 - 1.1.1 计算机系统的功能.....1
 - 1.1.2 计算机系统的组成.....2
 - 1.1.3 计算机系统的工作原理.....7
- 1.2 数据及其计算机内部表示.....8
 - 1.2.1 与数据相关的概念和术语.....8
 - 1.2.2 二进制表示法.....9
 - 1.2.3 文本数据表示法.....9
 - 1.2.4 数值数据表示法.....10
 - 1.2.5 音频数据表示法.....13
 - 1.2.6 图形和图像数据表示法.....13
 - 1.2.7 视频数据表示法.....14
 - 1.2.8 数据及其表示小结.....14
- 1.3 问题求解与算法.....15
 - 1.3.1 问题求解.....15
 - 1.3.2 算法及其特点.....15
 - 1.3.3 算法优劣的标准.....16
 - 1.3.4 算法描述.....16
- 1.4 程序设计语言与程序设计.....20
 - 1.4.1 程序设计语言的发展史.....21
 - 1.4.2 程序设计必备知识.....24
 - 1.4.3 结构化程序设计方法.....24
 - 1.4.4 程序质量.....26
- 1.5 计算机问题求解的过程.....26
 - 1.5.1 算法开发.....27
 - 1.5.2 算法实现.....27
- 习题1.....27

第2章 C语言概述.....29

- 2.1 C语言程序的组成及特点.....29
- 2.2 C语言程序上机指导.....30

- 2.3 C程序的调试.....32
- 习题2.....35

第3章 基本数据类型与数据运算.....36

- 3.1 基本标识符.....36
 - 3.1.1 保留关键字.....36
 - 3.1.2 预定义标识符.....37
 - 3.1.3 用户自定义标识符.....37
- 3.2 数据类型.....37
- 3.3 常量与变量.....39
 - 3.3.1 常量.....39
 - 3.3.2 变量.....43
- 3.4 基本数据类型的转换.....45
 - 3.4.1 自动类型转换.....46
 - 3.4.2 强制类型转换.....46
- 3.5 运算符和表达式.....47
 - 3.5.1 运算符和表达式概述.....47
 - 3.5.2 算术运算符与算术表达式.....48
 - 3.5.3 赋值运算符与赋值表达式.....50
 - 3.5.4 关系运算符与关系表达式.....52
 - 3.5.5 逻辑运算符与逻辑表达式.....52
 - 3.5.6 条件运算符与条件表达式.....54
 - 3.5.7 逗号运算符与逗号表达式.....54
 - 3.5.8 位运算符与位运算表达式.....55
 - 3.5.9 取长度运算符.....59
 - 3.5.10 运算符的优先级和结合性.....60
- 3.6 应用举例.....61
- 习题3.....63

第4章 常用库函数.....66

- 4.1 输出函数.....66

4.1.1	printf函数	66
4.1.2	putchar函数	69
4.1.3	puts函数	70
4.2	输入函数	70
4.2.1	scanf函数	70
4.2.2	getchar函数	72
4.2.3	gets 函数	72
4.3	字符串函数	73
4.3.1	strcat函数	73
4.3.2	strcpy函数	73
4.3.3	strcmp 函数	73
4.3.4	strlen函数	74
4.3.5	strlwr函数	74
4.3.6	strupr函数	74
4.4	数学函数	75
4.5	应用举例	77
	习题4	78

第5章 C程序结构及控制语句81

5.1	C程序语句及编程风格	81
5.1.1	C程序语句	81
5.1.2	编程风格	83
5.2	顺序结构	84
5.3	选择结构	85
5.3.1	if语句	85
5.3.2	switch语句	90
5.3.3	选择程序举例	92
5.4	循环结构	94
5.4.1	while语句	95
5.4.2	do-while语句	95
5.4.3	for语句	96
5.4.4	跳转语句	99
5.4.5	循环的嵌套	101
5.4.6	循环程序举例	103

5.5	应用举例	105
	习题5	108

第6章 数组 111

6.1	数组的定义和存储结构	111
6.1.1	数组的定义	111
6.1.2	数组的存储结构	112
6.2	一维数组	113
6.2.1	一维数组的定义	113
6.2.2	一维数组元素的引用	113
6.2.3	一维数组的初始化	114
6.2.4	一维数组程序举例	114
6.3	二维数组	119
6.3.1	二维数组的定义	119
6.3.2	二维数组元素的引用	119
6.3.3	二维数组的初始化	119
6.3.4	二维数组程序举例	120
6.4	字符串与字符数组	121
6.4.1	字符串	121
6.4.2	字符数组	122
6.4.3	字符串与字符数组程序举例	125
6.5	应用举例	127
	习题6	129

第7章 函数 131

7.1	引言	131
7.2	函数的定义	132
7.3	函数的调用	134
7.3.1	函数调用	134
7.3.2	函数的参数	135
7.3.3	函数的返回值	136
7.3.4	函数的声明	137
7.4	嵌套与递归	139
7.4.1	函数的嵌套调用	139
7.4.2	函数的递归调用	141

7.5 数组作为函数参数	145	9.3.2 指向数组元素的指针	193
7.5.1 数组元素作为函数参数	145	9.3.3 指向数组的指针	195
7.5.2 数组名作为函数参数	146	9.3.4 数组指针作参数	197
7.5.3 多维数组名作为函数参数	148	9.4 指针与字符串	201
7.6 变量的作用域与生存期	149	9.4.1 字符串的字符指针 表示形式	201
7.6.1 变量的作用域	149	9.4.2 利用字符指针访问字符串	202
7.6.2 变量的生存期	152	9.4.3 字符数组与字符指针的 比较	204
7.7 内部函数与外部函数	157	9.4.4 字符指针作函数参数	205
7.7.1 内部函数	157	9.5 指针数组	207
7.7.2 外部函数	158	9.5.1 一维指针数组的定义及使用	207
7.8 应用举例	158	9.5.2 指针数组作main函数的形参	208
习题7	165	9.6 指针与函数	209
第8章 预处理命令	170	9.6.1 返回指针值的函数	209
8.1 宏定义	170	9.6.2 指向函数的指针	212
8.1.1 无参数宏	170	9.6.3 函数指针作为函数的参数	214
8.1.2 带参数的宏	173	9.6.4 函数指针数组	215
8.2 文件包含	175	9.7 应用举例	217
8.3 条件编译	176	习题9	221
8.3.1 条件编译的格式	176	第10章 结构体与共用体	224
8.3.2 使用条件编译的优点	177	10.1 结构体	224
习题8	179	10.1.1 结构体类型的定义	224
第9章 指针	180	10.1.2 结构体变量的定义、引用 和初始化	225
9.1 引言	180	10.1.3 结构体程序举例	227
9.2 指针与指针变量	181	10.2 链表	232
9.2.1 指针的概念	181	10.2.1 链表概述	232
9.2.2 指针变量的定义及引用	182	10.2.2 静态链表	233
9.2.3 指针变量作为函数参数	186	10.2.3 动态链表	234
9.2.4 指向指针的指针	189	10.3 共用体	241
9.2.5 指针的运算	190	10.3.1 共用体的概念	241
9.2.6 void指针类型	191	10.3.2 共用体的定义	241
9.3 指针与数组	192		
9.3.1 数组的指针	192		

10.3.3 共用体变量的引用	242	11.7 应用举例	265
10.3.4 共用体应用举例	243	习题11	269
10.4 枚举类型	243	第12章 应用案例——学生宿舍	
10.4.1 枚举的概念	243	卫生管理系统	271
10.4.2 枚举的定义	244	12.1 需求陈述	271
10.4.3 枚举变量	245	12.2 需求分析	271
10.4.4 枚举应用举例	245	12.2.1 功能需求	271
10.5 应用举例	246	12.2.2 数据需求	272
习题10	250	12.2.3 技术约束	272
第11章 文件	252	12.3 总体设计	272
11.1 文件概述	252	12.3.1 系统总体结构	272
11.2 文件指针	253	12.3.2 全局数据结构	273
11.3 文件的打开与关闭	253	12.3.3 界面设计	273
11.3.1 文件的打开	254	12.4 详细设计	279
11.3.2 文件的关闭	255	12.4.1 系统主函数	280
11.4 文件的读/写	255	12.4.2 管理员部分	280
11.4.1 字符读/写函数	255	12.4.3 普通用户部分	287
11.4.2 字符串读/写函数	257	12.5 完整的代码实现与分析	287
11.4.3 字读/写函数	257	附录A 常用字符与ASCII码对照表 ..	314
11.4.4 数据块读/写函数	258	附录B C语言ANSI/ISO	
11.4.5 格式化读/写函数	259	标准库函数	316
11.5 文件的定位	261	附录C 实验	321
11.5.1 重返文件头函数	261	参考文献	328
11.5.2 指针位置移动函数	262		
11.5.3 取指针当前位置函数	263		
11.6 出错的检测	264		

第1章

绪 论

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的表示原理和方法；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的思想 and 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。

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，在各行各业都发挥着巨大的作用，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。在一般人看来，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。

1.1.1 计算机系统的功能

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，那么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。它主要具有数据输入、处理、存储和输出四大基本功能。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

要一系列指令的组合，所以程序是指令的集合。

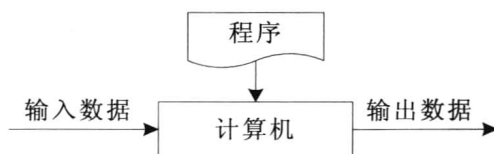


图 1.1 计算机功能视图

1.1.2 计算机系统的组成

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，是由硬件子系统和软件子系统两大部分组成的，这两部分相互联系、相互作用，共同实现计算机系统的功能。

1. 计算机硬件子系统

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四大基本功能，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五大基本部件构成，缺少其中任何一个部件就不能叫计算机了。

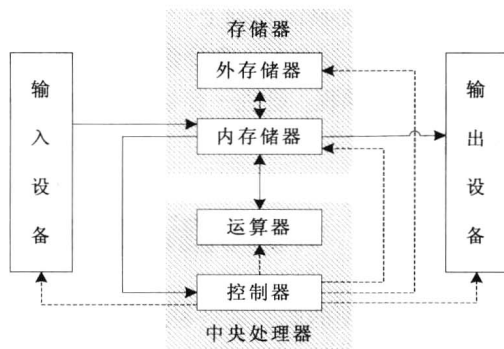


图 1.2 计算机硬件的基本组成

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。

(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。这些输入设备能够接收数字、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。在1.2节中将详细阐述各种类型的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。这些输出设备能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并输出。

由于输入/输出设备 (I/O设备) 都是机电、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) 为单位来计量, 其中, 1B (Byte) = 8b (Bit, 位), 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。存储器通常可分为三大类, 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。

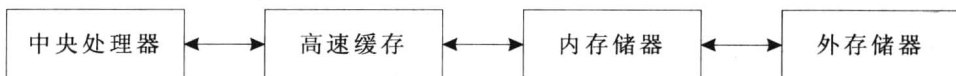


图 1.3 存储器的层次化结构

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, 在存储器层次化结构中离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, 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。其特点是比外存储器的存取速度快, 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, 断电后会丢失信息。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。利用这一特点常将基本输入/输出系统 (Basic Input Output System, BIOS) 固化在其中, 机器加电后立刻执行其中的程序。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00000H~FFFFFH, 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。

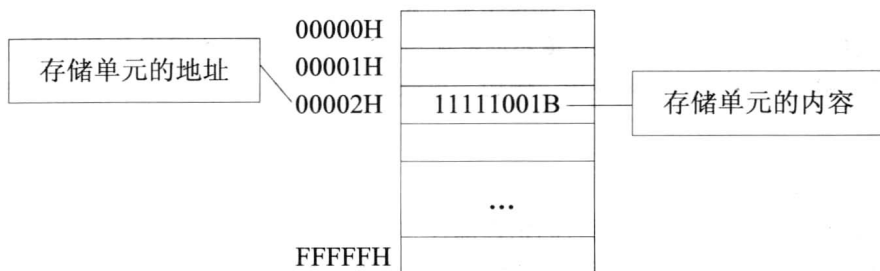


图 1.4 内存的结构

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长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，硬盘容量可达几百GB。外存可以长期保存大量程序和数据，是计算机中必不可少的重要设备。

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。当人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；同样地，当人们要读取外存的数据时，必须先按文件名找到指定的文件，然后从该文件中读取数据。

(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。

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，计算机就是在控制器的控制下有条不紊、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. 计算机软件子系统

(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中执行的程序。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中的表示、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, 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) 三大类。

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, 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, 但它很难在其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, 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, 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) 三大类。

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), 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公司的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

软件是软件工程师思维的产物，与硬件相比具有显著不同的特点，主要表现在以下几个方面。

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，而软件存在于软件工程师的脑中或纸面上，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了许多困难。

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及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。

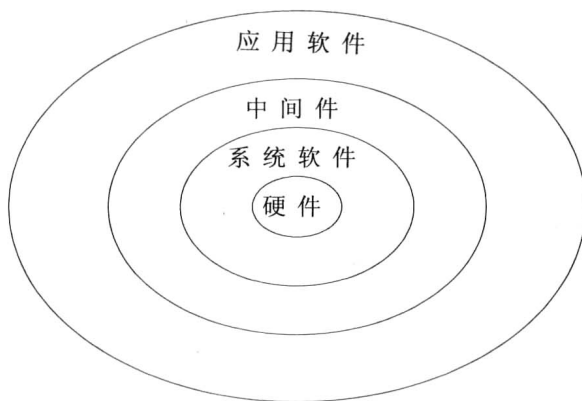


图 1.5 计算机系统层次结构

没有任何软件的计算机称为“裸机”，使用这样的机器必须通过改变计算机的硬件配置来执行不同的计算任务，这给计算机用户带来了很大的麻烦。增加了“系统软件”这一层以后，一般用户可以使用这些软件完成一些基本的功能操作，程序员可以使用这些软件所提供的API来编程，而不需要花费大量的精力去熟悉硬件。例如，一般用户可以使用Windows来完成文件复制、删除等各种文件管理操作，而程序员可以使用Windows API函数来编程。但系统软件对用户所提供的支持是有限的，由此，需要更多的面向特定领域、解决特定问题的应用软件。有了这些应用软件，计算机的一般用户就可以更加方便、有效地使用计算机，例如，财务管理人员可以使用财务管理软件来完成复杂的财务管理任务，而程序员的任务是编出各种应用软件的

程序代码 (Code)。

计算机软件 and 硬件是相互联系、相互作用的。软件依赖于硬件这一物质条件，软件中所定义的各种功能最终要依赖硬件来完成，没有硬件的支持，软件也就失去了其运行的基本环境；而硬件需在软件的支配下才能有效地工作。有了软件，用户面对的将不再是物理计算机，而是一台抽象的逻辑计算机。人们根本不必了解计算机的硬件细节，就能通过不同的软件方便、高效地使用计算机完成各种工作。从这个意义上说，软件是用户与计算机的接口。通常情况下，人们是通过软件来使用计算机系统的。

1.1.3 计算机系统的工作原理

计算机的工作过程就是执行程序的过程。怎样组织程序，涉及计算机体系结构问题。现在的计算机都是基于“程序存储”概念设计制造出来的。

1. “程序存储”的概念

冯·诺依曼 (Von Neumann) 是美籍匈牙利数学家，他在1946年提出了关于计算机组成和工作方式的基本设想。冯·诺依曼设计思想可以简要地概括为以下3点。

- 计算机应包括运算器、存储器、控制器、输入设备和输出设备五大基本部件。
- 在计算机内部，指令和数据都以二进制模式存储到存储器中。
- 每个程序由一组有限的指令序列组成，在控制器的控制下，程序无须操作人员干预就能自动逐条执行预先定义好的指令。

冯·诺依曼设计思想最重要之处在于明确地提出了“程序存储”的概念，他的全部设计思想实际上是对程序存储概念的具体化。到现在为止，尽管计算机制造技术已经发生了极大的变化，但是就其体系结构而言，仍然是根据他的设计思想制造的，这样的计算机称为冯·诺依曼结构计算机。

2. 计算机的工作过程

如果能让计算机完成某项计算任务，就得先把程序编出来，并通过输入设备送到存储器中保存起来，接下来就是执行程序的问题了。计算机的工作过程就是执行程序的过程。根据冯·诺依曼的设计，计算机应能自动执行程序，程序执行的过程可以概括为以下5步。

(1) 读取一条指令：程序计数器 (PC) 是一个特殊的寄存器，它用来存放下一条要执行的指令的地址。控制器访问PC中指定的内存地址，将其内容送入指令寄存器，更新PC的内容使其指向下一条要执行的指令的地址。

(2) 译解指令：控制器负责理解指令寄存器中指令的含义，并将其译解成相应的控制信号。

(3) 如果需要，获取数据：如果被执行指令的操作数存储在内存中，则需要额外的内存访问以获得执行指令所必要的操作数。

(4) 执行指令：控制器根据译解的结果向各个部件发出相应的控制信号，完成指令规定的操作。

(5) 读取下一条指令，转至 (2)，直到程序所有的指令执行完毕。

1.2 数据及其计算机内部表示

作为一个通用的数据处理器，没有数据，计算机就毫无用处。因此，用适当的方式表示和组织数据是非常重要的。早期的计算机，由于性能方面的原因，所能处理的数据仅仅是数字和文本，但现在的计算机已经成为真正的多媒体（Multimedia）处理设备，可以处理数字（Digit）、文本（Text）、音频（Audio）、视频（Video）、图形（Graphics）、图像（Image）等多种类型的数据。例如，可以使用播放器来播放音乐和电影，使用各种图形图像处理软件来生成、存储和处理图形图像。

1.2.1 与数据相关的概念和术语

数据是对客观事物的符号表示，在计算机科学中是指所有能输入到计算机中并被计算机程序处理的符号的总称。

（1）数据元素（Data Element）：数据元素是对某种事物一个具体实例各方面属性的符号化描述结果，在计算机程序中通常作为一个整体进行考虑和处理，是数据的基本单位。在不同的应用背景下，也把数据元素称为节点、记录或条目等。如表1.1所示的学生C语言成绩表，其中一条学生的C语言成绩信息就是一个数据元素。

表1.1 学生C语言成绩表

学号	姓名	班级	C语言成绩	位次
200605101101	李莉	软件工程05-1班	96	1
200605101202	张三	软件工程05-2班	85	10
...	...	...	...	...

（2）数据项（Data Item）：数据项是具有独立意义的不可分割的最小数据单位，是对数据元素一个属性的描述。数据项也被称为域或字段。一个数据元素可由多个数据项组成。例如，一条学生的C语言成绩信息由学号、姓名、班级、C语言成绩和位次等数据项组成。

（3）数据对象（Data Object）：数据对象是性质相同的数据元素的集合，是数据的一个子集。例如，所有英文字母构成的集合是一个字母数据对象；所有学生C语言成绩构成的集合是学生C语言成绩数据对象。

（4）数据结构（Data Structure）：计算机加工处理的数据元素不是相互孤立的，它们之间应当存在某些逻辑上的关联，这种数据元素间的关联关系称为结构。数据元素间通常包含集合、线性、树形和图状（网状）4种基本结构。所谓集合结构，是指结构中的数据元素同属于一个集合；线性结构是指结构中的数据元素存在一对一的关系；树形结构是指结构中的数据元素之间存在一对多的关系；图状或网状结构是指结构中的数据元素之间存在多对多的关系。数据结构是存在一种或多种特定关系的数据元素的集合，其外在表现为数据的组织形式。

（5）数据类型（Data Type）：数据类型用于描述程序处理对象的特性。在用高级语言编写程序时，每个变量、常量或表达式都有一个确定的数据类型。数据类型明显或隐含地规定了在程序执行期间变量或表达式的取值范围，以及在這些值上允许进行的操作。因此，数据类型是一个