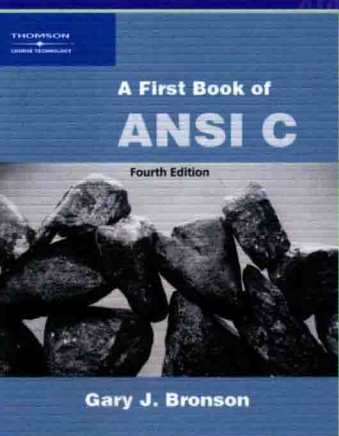


# 标准C语言基础教程 (第四版)

## A First Book of ANSI C Fourth Edition



[美] Gary J. Bronson 著  
张永健 等译

国外计算机科学教材系列

# 标准 C 语言基础教程

(第四版)

A First Book of ANSI C

Fourth Edition

[美] Gary J. Bronson 著

张永健 等译

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

这是一本介绍用 C 语言进行计算机编程的经典教材。通过大量的实例和练习,全书系统介绍了数据类型、算术运算、逻辑运算、变量、条件语句、函数、数组、指针、字符串、结构、文件操作、位操作、宏、库函数等基本内容,使读者在阅读之后就能很快掌握 C 语言编程的精髓。讲解 C++ 编程的一章也是本书的特色之一。书中每章都有大量的简答题和编程练习题,附录还列出了它们的答案。

本书结构合理,内容深入浅出,适合作为高等学校相关专业的本科和专科教材,也适合初学编程的自学者阅读。

A First Book of ANSI C, Fourth Edition

Gary J. Bronson 张永健

Copyright © 2006 Cengage Learning Asia Pte Ltd.

Original edition published by Cengage Learning. All Rights reserved.

Publishing House of Electronics Industry is authorized by Cengage Learning to publish and distribute exclusively this simplified Chinese edition. This edition is authorized for sale in China Mainland. Unauthorized export of this edition is a violation of the Copyright Act. No part of this publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

本书原版由圣智学习出版公司出版。版权所有,盗印必究。

本书中文简体字翻译版由圣智学习出版公司授权电子工业出版社独家出版发行。此版本经授权仅限在中国大陆销售。未经授权的本书出口将被视为违反版权法的行为。未经出版者预先书面许可,不得以任何方式复制或发行本书的任何部分。

本书封面贴有 Cengage Learning 防伪标签,无标签者不得销售。

版权贸易合同登记号 图字:01-2006-5601

## 图书在版编目(CIP)数据

标准 C 语言基础教程/(美)盖瑞·J. 布朗森(Gary J. Bronson)著;张永健等译。—4 版.

北京:电子工业出版社,2018.1

(国外计算机科学教材系列)

书名原文:A First Book of ANSI C, Fourth Edition

ISBN 978-7-121-33272-2

I. ①标… II. ①盖… ②张… 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(2017)第 308655 号

策划编辑:冯小贝

责任编辑:冯小贝

印 刷:三河市鑫金马印装有限公司

装 订:三河市鑫金马印装有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1092 1/16 印张:37.25 字数:960 千字

版 次:2018 年 1 月第 1 版(原著第 4 版)

印 次:2018 年 1 月第 1 次印刷

定 价:89.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010)88254888,88258888。

质量投诉请发邮件至 zltz@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式:fengxiaobei@phei.com.cn。

## 译者序

本书是 Gary J. Bronson 所著的 A First Book of ANSI C, Fourth Edition 的中译本,作者以相当清晰的 C 语言程序设计概念精辟地介绍了 C 语言的基础知识和基本结构,并以丰富的 C 语言编程经验循序渐进地引导学生进入神秘的 C 语言程序设计的殿堂。新版本中提供了更为完善的 ANSI C 语言的学习环境和更丰富的专业内容。

实践证明,在正确牢固地掌握了 ANSI C 语言的基础理论知识以后,进一步学习 C++ 语言、Java 语言等就不再是困难的事情。而入门是最重要的第一步,这本书为学生以及初学编程者提供了学习 C 语言的全面而准确的讲解及指导,特别是对学习 Windows 和 UNIX/Linux 环境下的 C 语言程序设计的学生而言,本书不失为难得的一本好书。

全书由国际关系学院张永健主持翻译并统筹专业术语的译法。翻译人员的具体分工如下。前言、第 1 章至第 4 章由张永健翻译,第 5 章由赵璐翻译,第 6 章和第 7 章由常征翻译,第 8 章由经乃鹏翻译,第 9 章由龙继文翻译,第 10 章至第 12 章由胡志强翻译,第 13 章由颜焯翻译,第 14 章由胡彦平翻译,第 15 章由鲁敏翻译,附录及封底由张思宇翻译。

在翻译过程中,参考了本书中文版 2006 年版本中的一些译法,并订正了其中的一些错误。译者对空军航空维修技术学院单先余老师所做的前期工作表示衷心感谢。

由于本书的翻译时间紧,且是多人共同完成的,译稿中如有不妥之处,欢迎广大读者批评指正。

译者

# 前 言

正如本书的前几版一样,这一版本的主要目的是使 C 语言成为一种可利用的应用程序编写语言。本书的前几版取得了成功,而且从学生和那里获得的评论来看,表明本书确实能够帮助他们学习和讲授 C 语言,这确实令人非常满意。基于此,第四版的目标依然与前三版相同:清晰地、无歧义地向初学 C 语言的学生以可以理解的方式讲解所有的主题。本书可以作为一般性的编程导论,尤其适合作为 C 语言的入门性教材,还可以作为进一步学习 C++ 语言的基础。

这个版本中对许多内容进行了加强,包括如下这些:

- 展示真实的编程应用的新案例研究。
- 对输入数据验证技术的透彻解释。
- 对练习进行了扩充,使其既包含简答题也有编程练习。
- 大多数章节的末尾都包含一个“常见编译错误”表,它既针对 UNIX 编译器也针对 Windows 编译器。此外,多数章节的末尾还列出了常见的编程错误。
- 与计算机科学主题相关的历史注解。
- 增加了一章,介绍与计算机硬件和软件工程有关的内容。
- 将所有的程序和描述更新成满足 C99 ANSI 标准。

## 本书的独到之处

**突出重点** 本书在给出主题时采用的方式是让学生对现实世界的问题进行编程。这个特点被一位评审人员贴切地描述了,他写道:(本书的)深度在于使初学者感到困惑的那些问题上,这与那些只提供大量内容而没有包含有用的提示和捷径的(许多)教材形成了鲜明对比。

**写作风格** 我坚信,不是入门性的教材在教授学生知识——教授学生是老师的责任。作为一本有用的入门性教材,充当的一定是对起主导作用的老师的支持性角色。但是,只要老师搭建好了舞台,教材就必须鼓励学生掌握课堂中讲解过的材料。为此,教材对学生来说必须是表述清楚的。我所主要关注的,同时也是本书与众不同的特点之一,即它是为学生而写的。正如一位评审人员对本书前一个版本所说的:本书关注的是学生而不是专业人员。

**软件工程** 从一开始,本书就向学生介绍了软件工程的基础知识。1.3 节中将讲到算法以及用来描述算法的各种方法。1.4 节中会继续讲解软件工程,这一节是介绍软件开发过程的。在后续所有演示软件开发过程实际应用的案例研究中,同样强调了软件工程。

**指针介绍** 本书第一版的一个特点是对指针的介绍,书中采用的是先用 `printf()` 函数显示变量的地址,然后才介绍用于实际存储这些地址的变量。与第一版出版时流行的间接描述方法相比,这种方法似乎更符合逻辑,它是一种解释指针变量的直观方法。从第一版开始,我高兴地看到使用 `printf()` 函数显示地址已经成为了介绍指针的标准方法。因此,尽管这种方法不再是本书独有的特点,但我仍对这种处理方式感到自豪,并将在这个新版本中继续使用它。

**程序测试** 教材中的每一个 C 语言程序都已经用 Microsoft 的 Visual C++ .NET 和 UNIX 编



译器成功编译并运行过,所有的程序都遵循 C99 ANSI 标准。书中使用的全部程序例题的源代码文件,都可以通过本书的配套网站免费获得。这样做就使学生可以用这些程序体验编程,并使学生能更容易地根据各节后面的练习的要求修改它们。

## 教学特点

为了使这门课程达到一流教学水平,本书具备如下的教学特点。

**节后练习** 除了前几个版本中提供的编程练习之外,书中的每一节几乎都包含大量的简答题。此外,所有简答题的答案都在附录 G 中给出,而所有编程练习的答案可由教师从本书的配套网站或出版社获得。

**伪代码和流程图描述** 伪代码在本书中处处被强调。学生还要学习流程图符号以及如何使用流程图来可视化地表示流程控制结构。

**常见编程和编译器错误以及各章小结** 每一章都用“编程和编译器错误”一节结尾。这一版本中新增加的是容易阅读的一个表,它的内容是由 UNIX 编译器和 Windows 编译器产生的编译器错误以及相关的错误消息。每一章中还包含了一个本章中涵盖的主要主题的小结。

**编程注解和历史注解** 散布在各章中标有“编程注解”的方框,强调的是重要的概念、有用的技术点以及专业程序员所使用的编程技巧。同样,“历史注解”强调的是与计算机硬件和软件发展史相关的重大历史事件和人物。

**附录和答案** 这一版本中对附录进行了扩充。这些附录中讲解的是运算符优先级、ASCII 码、标准 C 函数库、I/O 与标准错误重定向、浮点型数字存储以及创建个人 C 函数库。最后一个附录中给出的是全部简答题的答案。全部编程练习的答案可从网站 [www.course.com](http://www.course.com) 获得。

## 教辅资料

本书用于课堂教学时可获得如下教辅资料(教师可联系 [Te\\_service@phei.com.cn](mailto:Te_service@phei.com.cn) 申请教辅资料)。

**电子版本的教师手册** 随本书提供的教师手册包含如下内容:

- 额外的帮助备课的指导材料,包括讲授主题的建议。
- 每章后面的所有问题的答案,包括编程练习的答案。

**ExamView** 本书与 ExamView 软件相配套。ExamView 是一个功能强大的考试软件包,允许教师创建并执行纸质试卷考试、(基于局域网的)计算机考试以及通过因特网进行的考试。ExamView 包含数百个与本书所涵盖的主题相关的问题,能为学生建立详细的学习指南。这些基于计算机和因特网的测试组件,使学生能在自己的计算机上进行考试,并能够节省教师的时间,因为每次考试都是自动评分的。

**PowerPoint 演示** 本教材的每一章都提供 PowerPoint 幻灯片。这些幻灯片是课堂演示的教学辅助手段,教师可以从网络上下载它们用于章节复习,也可以将它们打印出来发给学生。教师也可以为课堂上讲解的其他主题增加自己的幻灯片。

**远程学习** Course Technology 公司通过 WebCT 和 Blackboard 提供在线课程,提供最全面的动态学习体验。将在线内容添加到某个课程中时,会增加许多内容:主题回顾、实践测试、复习题、作业、PowerPoint 演示等,而最重要的是,它是通向 21 世纪最重要的信息资源的大门。关于如何将远程学习引入课程的更多信息,请咨询当地的 Course Technology 公司的销售代表。

**源代码** 书中所用的源代码可从网站 [www.course.com](http://www.course.com) 获得。

**答案文件** 书中所有编程练习的答案可从网站 [www.course.com](http://www.course.com) 获得。

## 致谢

这一版本是过去几个成功版本的直接结果。在这一点上,最诚挚的感谢和感激要献给那些发现这几个版本对他们的 C 语言教学和学习有所帮助的教师和学生。

还要特别感谢 Course Technology 公司的编辑 Alyssa Pratt。他的眼光、一贯的信任以及对进度和细节的关注,使得本书得以成功完成。接下来,要感谢开发编辑 Ann Shaffer,是她对原始手稿进行了最广泛、最专业的编辑,能得到这样的编辑和建议是一件幸事。

此外,还要向下列评审人员表达谢意:

Saint Rose 学院的 John Avitabile

Colorado 大学 Colorado Springs 校区的 Pamela Carter

Hudson Valley 社区学院的 Andrew Hurd

Colorado 大学 Denver 校区的 Thami Rachidi

Colorado 大学 Denver 校区的 Eric Thompson

Blinn 学院的 John H. Town

每一位评审人员都对本书提出了详细和建设性的意见。在从原稿成型到成熟的整个编辑过程中,他们的建议、对细节的关注以及意见,都非常有帮助。

一旦评审过程完成,把最终手稿变成教材的任务,就取决于许多人而不是我一个人了。特别感谢产品编辑 Jennifer Roehrig、质量保证测试员 Serge Palladino、质量保证经理 Chris Scriver 以及完成了本书练习题答案的 Nicole Ashton,并要再次感谢负责协调整体事务的 Ann Shaffer。这个团队的贡献对我来说是难以置信而又非常重要的,非常感谢为这本书工作过的每一个人。

特别的感谢要献给为本书提供过材料的三位同事。首先,除了 Hudson Valley 社区学院的助理教授 Andrew J. Hurd 提供的大量编辑和技术的贡献之外,还要感谢他提供的编译器错误材料。非常感谢已从美国 Weber 州立大学退休的 R. Kenneth Walter,他愉快地提供了书中“历史注解”部分所使用的材料。特别的感谢还要献给我的第一位数学老师 Marie Scully-Bell,她教会我不管遇到多么难的问题,都一定能够克服,不管这样的难题是学术上的还是生活上的。她是我们有幸能在生命中遇到的特别的人之一。一如既往,书中的任何错误(就像我生活中的错误一样)都由我一人承担。

还要衷心感谢 Fairleigh Dickinson 大学的直接鼓励和支持。这包括不断的鼓励、支持,以及由 Kenneth Greene 博士和 Paul Yoon 博士提供的良好学术氛围。没有他们的支持,这本教材不可能写成。

最后,要深深地感激我的朋友、妻子和亲密的伙伴 Rochelle 给予的耐心、包容和爱。

致:Rochelle, Matthew, Jeremy, David Bronson

Gary Bronson

2006

# 目 录

## 第一部分 基础知识

第1章 计算机编程导论 .....	2
1.1 历史和硬件 .....	2
1.2 编程语言 .....	8
1.3 算法 .....	13
1.4 软件开发过程 .....	17
1.5 案例研究:设计与开发 .....	24
1.6 编程错误 .....	28
1.7 小结 .....	28
1.8 补充材料:数字存储码 .....	29
第2章 C语言编程初步 .....	31
2.1 C语言编程简介 .....	31
2.2 编程风格 .....	40
2.3 数据类型 .....	43
2.4 算术运算 .....	49
2.5 变量和声明 .....	58
2.6 案例研究:温度转换 .....	66
2.7 编程错误和编译器错误 .....	70
2.8 小结 .....	72
2.9 补充材料:内存分配 .....	73
第3章 数据处理与交互式输入 .....	79
3.1 赋值 .....	79
3.2 数学库函数 .....	88
3.3 交互式输入 .....	92
3.4 格式化输出 .....	102
3.5 符号常量 .....	110
3.6 案例研究:交互式输入 .....	112
3.7 编程错误和编译器错误 .....	117
3.8 小结 .....	119
3.9 补充材料:抽象简介 .....	120

## 第二部分 控制流

第4章 选择 .....	124
4.1 关系表达式 .....	124



4.2	if 语句和 if...else 语句 .....	129
4.3	if...else 链 .....	139
4.4	switch 语句 .....	146
4.5	案例研究:数据验证 .....	152
4.6	编程错误和编译器错误 .....	156
4.7	小结 .....	158
4.8	补充材料:错误,测试和调试 .....	160
<b>第 5 章</b>	<b>循环 .....</b>	<b>165</b>
5.1	基本的循环结构 .....	165
5.2	while 语句 .....	167
5.3	利用 while 循环求和及平均值 .....	174
5.4	for 语句 .....	184
5.5	案例研究:循环编程技术 .....	192
5.6	嵌套循环 .....	199
5.7	do...while 语句 .....	202
5.8	编程错误和编译器错误 .....	205
5.9	小结 .....	206
<b>第 6 章</b>	<b>函数模块性(1) .....</b>	<b>209</b>
6.1	函数声明与参数声明 .....	209
6.2	返回值 .....	221
6.3	案例研究:计算年龄标准 .....	229
6.4	标准库函数 .....	238
6.5	编程错误和编译器错误 .....	250
6.6	小结 .....	251
<b>第 7 章</b>	<b>函数模块性(2) .....</b>	<b>252</b>
7.1	变量的作用域 .....	253
7.2	变量存储类 .....	258
7.3	按引用传递 .....	264
7.4	案例研究:交换值 .....	275
7.5	递归 .....	281
7.6	编程错误和编译器错误 .....	286
7.7	小结 .....	287
<b>第三部分 基础知识补充</b>		
<b>第 8 章</b>	<b>数组 .....</b>	<b>290</b>
8.1	一维数组 .....	290
8.2	数组初始化 .....	298
8.3	数组作为函数实参 .....	302
8.4	案例研究:计算平均值和标准差 .....	307

8.5	二维数组 .....	312
8.6	编程错误和编译器错误 .....	320
8.7	小结 .....	321
8.8	补充材料:查找和排序方法 .....	322
<b>第9章</b>	<b>字符串 .....</b>	<b>342</b>
9.1	字符串基础 .....	342
9.2	库函数 .....	352
9.3	输入数据验证 .....	358
9.4	格式化字符串 .....	364
9.5	案例研究:字符和单词计数 .....	367
9.6	编程错误和编译器错误 .....	372
9.7	小结 .....	373
<b>第10章</b>	<b>数据文件 .....</b>	<b>375</b>
10.1	声明,打开和关闭文件流 .....	375
10.2	读取和写入文本文件 .....	385
10.3	随机文件访问 .....	393
10.4	传递和返回文件名 .....	396
10.5	案例研究:创建和使用常量表 .....	399
10.6	写入和读取二进制文件 .....	408
10.7	编程错误和编译器错误 .....	412
10.8	小结 .....	413
10.9	补充材料:控制码 .....	414

#### 第四部分 其他主题

<b>第11章</b>	<b>数组,地址和指针 .....</b>	<b>418</b>
11.1	数组名称作为指针 .....	418
11.2	指针操作 .....	423
11.3	传递和使用数组地址 .....	428
11.4	使用指针处理字符串 .....	435
11.5	使用指针创建字符串 .....	439
11.6	编程错误和编译器错误 .....	444
11.7	小结 .....	446
<b>第12章</b>	<b>结构 .....</b>	<b>447</b>
12.1	单一结构 .....	447
12.2	结构数组 .....	452
12.3	传递结构和返回结构 .....	456
12.4	联合 .....	463
12.5	编程错误和编译器错误 .....	465
12.6	小结 .....	466

第 13 章 动态数据结构 .....	467
13.1 链表简介 .....	467
13.2 动态内存分配 .....	474
13.3 栈 .....	479
13.4 队列 .....	485
13.5 动态链表 .....	491
13.6 编程错误和编译器错误 .....	497
13.7 小结 .....	499
第 14 章 其他功能 .....	501
14.1 新增的特性 .....	501
14.2 按位运算 .....	506
14.3 宏 .....	513
14.4 命令行参数 .....	515
14.5 编程错误和编译器错误 .....	519
14.6 小结 .....	520
第 15 章 C++ 简介 .....	521
15.1 C++ 中的过程化编程 .....	521
15.2 面向对象的 C++ .....	526
15.3 编程错误和编译器错误 .....	527
15.4 小结 .....	528
附录 A 运算符优先级表 .....	530
附录 B ASCII 字符码 .....	531
附录 C 标准 C 语言库 .....	534
附录 D 输入, 输出和标准错误重定向 .....	538
附录 E 浮点数存储 .....	540
附录 F 创建个人函数库 .....	542
附录 G 简答题答案 .....	543

总功耗为18 000个真空电子管,重量约为30 100千克,每秒能进行200次加法或300次乘法(见图1.3)。

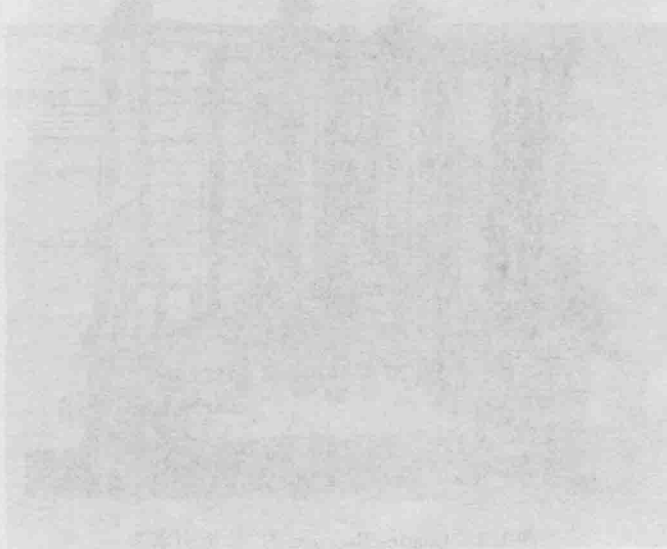
# 第一部分

## 基础知识

### 第1章 计算机编程导论

### 第2章 C语言编程初步

### 第3章 数据处理与交互式输入



之需求的分工更重,甚至更加复杂而精密,大大提升五二零的运算速度。1949年,美国麻省理工学院研制成功第一台电子计算机ENIAC,它由1.8万个电子管组成,重量达30吨,功耗为18千瓦,每秒能进行200次加法或300次乘法。ENIAC的研制成功标志着电子计算机时代的开始。

# 第 1 章 计算机编程导论

## 1.1 历史和硬件

用机器执行计算的过程，几乎与有记录的历史一样古老。最早的计算工具是算盘，在当今的中国仍很常见，它就如同美国的手持式计算器。但是，这两类设备都需要人的直接参与才能使用。在算盘上将两个数相加要求拨动算珠，用计算器相加两个数则要求按下两个数字键和加号键。

据记载，建造一台可编程的计算机器的首次尝试，是在 1822 年由英国的 Charles Babbage 进行的。诗人 Lord Byron 的女儿 Ada Byron 开发出了一个指令集，可用于操作这台机器(实际上这台机器没有建造出来)。尽管这台被 Babbage 称为“分析引擎”的机器在他的有生之年没有被成功地建造出来，但是“可编程的机器”的概念得以保留。1937 年，美国爱荷华州立大学的 John V. Atanasoff 博士和一位姓名为 Clifford Berry 的研究生利用电子元器件部分地实现了这种机器(见图 1.1)。这台机器被称为 ABC (Atanasoff-Berry Computer)，运行时要求一位操作员控制外部的配线以执行预期的运算。因此，内部存储一组可替代的指令的目标还是没有达到。

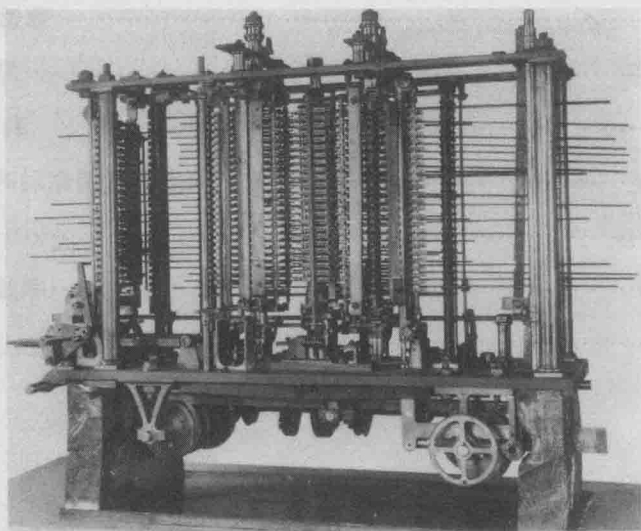


图 1.1 Charles Babbage 的“分析引擎”

1939 年末爆发的第二次世界大战，使得开发计算机的需求更加强劲。这项工作的先驱之一是美国宾夕法尼亚大学摩尔工程学院的 John W. Mauchly 博士，他曾经拜访过 Atanasoff 博士并见过他的 ABC 机器。后来，Mauchly 博士就着手与 J. Presper Eckert 一起研制一种称为 ENIAC (Electrical Numerical Integrator and Computer) 的计算机。这个项目的资金由美国政府提供。这台机器最早实现的功能之一是计算大炮发射的弹道的轨迹。到 1946 年完成时，ENIAC



已经包含 18 000 个真空电子管, 质量约为 30 000 千克, 每秒能执行 5000 次加法或 360 次乘法(见图 1.2)。

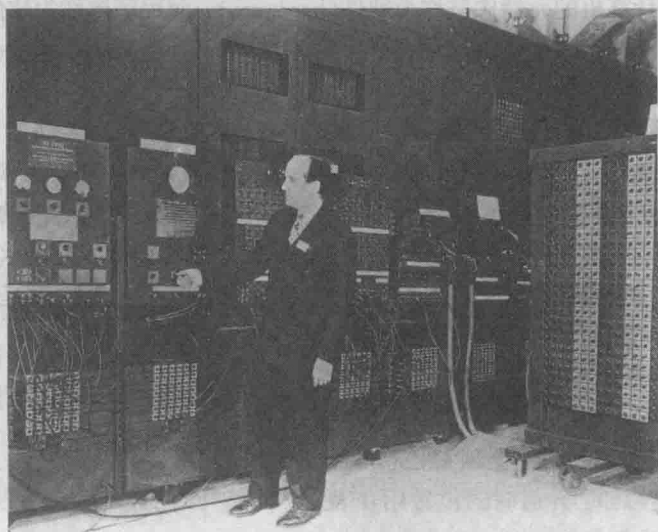


图 1.2 ENIAC

正当使用真空电子管的 ENIAC 的研制工作正在进行时, 哈佛大学已经在从事利用机械继电器开关研制名为 Mark I 的计算机(见图 1.3)。Mark I 于 1944 年完成, 不过它每秒只能执行 6 次乘法。这两台机器与 ABC 一样, 都需要外部连线才能执行预期的运算。



图 1.3 Mark I

1949 年 5 月 6 日, 随着剑桥大学 EDSAC(Electronic Delayed Storage Automatic Computer, 电子延时存储自动计算机)的成功运行, 能在机器内部存储指令和数据的计算机的目标终于实现了。除了能够执行计算外, EDSAC 还能够存储数据和指导计算机操作的指令。EDSAC 结合了一种存储器的形式, 存储器的原理由 John Von Neumann 提出, 它允许取得(retrieve)一条指令, 然后取得

执行该指令所需的数据。现在,相同的设计和操作原理仍然被大多数计算机所使用。唯一有显著改变的特点是用于制造计算机的元器件的大小和执行速度,以及存储在计算机中的程序类型。就整体而言,用于制造计算机的元器件被称为硬件(hardware),而程序被称为软件(software)。

## 计算机硬件

计算机由称为硬件的物理元器件构造。硬件的用途是在一个已存储的程序的指导下使数据的存储和处理更方便。如果计算机硬件能够和人一样使用相同的符号存储数据,则数字 126 就可以用符号 1, 2 和 6 存储。类似地,我们称为“A”的字母也可以用相同的符号存储在计算机硬件中。遗憾的是,计算机的内部器件要求一种不同形式的数字和字母表示方法。有必要先理解为什么计算机不能使用我们的符号之后,这样才能理解数字在机器内部是如何表示的。这将使理解用于存储和处理数据的计算机的实际部件更容易一些。

## 位与字节

计算机中最小和最基本的数据项是位(bit)。从物理上看,一个位实际上就是一个能开启或断开的开关。按照习惯,开启时的位置用 0 表示,而断开时的位置用 1 表示<sup>①</sup>。

### 历史注解:二进制的 ABC

在几年的时间里,Iowa 州立大学的 Atanasoff 博士都在为如何设计一台能帮助他的研究生解决复杂的方程式求解问题的计算机而大伤脑筋。他考虑建造一台基于二进制数的机器,二进制数是用于具有两个容易辨别状态(开和关)的机电装置最自然的系统,但他担心人们不会使用这种不是基于熟悉和方便的十进制系统的机器。

最后,1937 年在伊利诺伊州一家小旅馆里一个寒冷的夜晚,他终于下决心采用最简单且成本最小的二进制数(位)来制造他的计算机。接下来的两年里,他和研究生 Clifford Berry 建造出了第一台电子数字计算机,称为 ABC(Atanasoff-Berry Computer)。从那以后,绝大多数的计算机就是二进制机器了。

单一的位只能表示值 0 和 1,这导致了位的可用性有限。因此,为了存储和传输数据,所有的计算机都将一定数量的位分成组。8 个位的组合可以形成一个更大的单元,它是一个几乎通用的计算机标准,称为 1 字节(byte)。1 字节的 8 个位中的每一个位,可以为 0 或 1,总共能够表示 256 种不同的模式(pattern)。这些模式由 00000000(8 个开关全开)到 11111111(8 个开关全关)以及二者之间所有的 0 和 1 的可能组合构成。每一个这样的模式,都能够用来表示字母表中的 1 个字母、单个的字符(美元符、逗号等)、1 位数字或者多位数字。由 0 和 1 组成的用来表示字母、数字以及其他单字符的模式集合,被称为字符码(字符码中的 ASCII 码将在 2.1 节中介绍)。

对于类似名称和地址以及任何需要处理的文本而言,字符码是极其有用的。但是,对于算术数据,它几乎从来没有被使用过。这有两方面的原因。首先,将十进制数转换成字符码需要为每个数字准备一个代码。对于大数字,这会浪费计算机的内存空间。然而更基本的原因是:基于数字 10 的十进制计数系统本身就不被计算机内部的硬件所支持。前面说过,位是计算机的基本存储部件,它只能具备两种可能的状态之一,分别表示 0 和 1。这表明使用基于这两个状态的计数系统将更具意义,而事实也是如此。1.8 节中将讲解最常使用的基于 2 的计数系统。

<sup>①</sup> 遗憾的是,这个习惯有点武断,并且你常常会遇到相反的情况,开和关的位置分别表示为 1 和 0。

对于计算机内部的计数系统与十进制系统不同的情况,我们应该不会感到奇怪。例如,你也许已经熟悉了下面这两个计数系统,并且能够容易地识别它们。

- 罗马数字: XIV
- 哈希标记系统: ### ### ////

## 部件

所有的计算机,从价值数百万美元的大型超级计算机到价值几百美元的小型桌面个人计算机,都必须执行一个最小的任务集,并具备完成下列任务的能力:

1. 接受输入,包括数据和指令。
2. 显示输出,包括文本和数字。
3. 存储数据和指令。
4. 对输入的数据或已存储的数据执行算术和逻辑运算。
5. 监督、控制和指导系统的全部操作和顺序。

图 1.4 中给出的是支持这些能力以及共同组成计算机硬件的那些计算机部件。

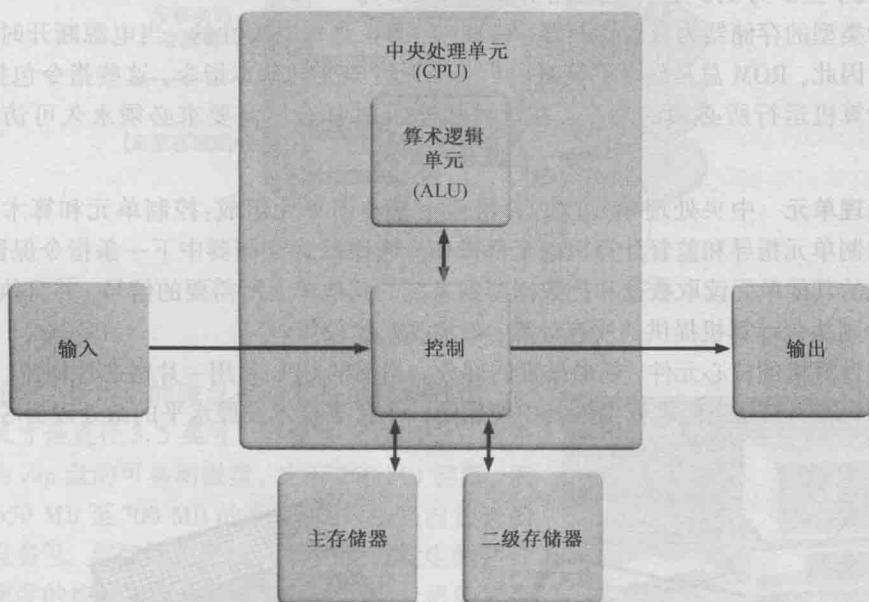


图 1.4 计算机的基本硬件单元

### 历史注解:图灵机

20 世纪 30 年代和 40 年代, Alan Mathison Turing(1912–1954) 和其他人提出了一种描述计算机应该能够做什么的理论。他的理论机器被称为图灵机,这台机器包含解决编程问题的最小操作集。通过给这样一台假设的计算机一组指令, Turing 希望能够证明所有的问题都可以解决。在证明的过程中获得的成果是:有些问题不能被任何机器解决,就像有些问题不能被人解决一样。在第一台电子计算机建造之前, Turing 的工作就奠定了计算机的理论基础。对于二战期间开发出了关键的密码破译计算机的团队而言, Turing 的贡献是使他的理论得以具体实现。

**主存储器单元** 这个单元按照字节顺序存储数据和指令。如果某个程序要操作计算机,则它必须驻留在主存储器中。主存储器将 1 个或多个字节组合成一个单元,称为字(word)。尽管更大尺寸的字有利于计算机速度和能力的提升,但这种提升是通过增加计算机的复杂性达到的。

早期的个人计算机(PC),例如 Apple IIe 和 Commodore 计算机,在内部存储和传送由 1 字节组成的字。第一台 IBM PC 使用由 2 字节组成的字,而目前更多的基于 Intel 芯片的 PC 存储和处理由 4 字节组成的字。

字在计算机存储器中的排列类似于大旅馆中套间的排列,旅馆中的每个套间都由大小相同的房间组成。就像每个套间都有唯一的房号让顾客查找和确定它一样,计算机存储器中每个字也都有唯一的数字地址。与房号一样,字地址总是正的、无符号的、用于查找和确定用途的完整数字。此外,就像旅馆房间可以通过房门连接形成更大的套间一样,字也能够组合成更大的单元,以适应不同大小的数据类型。

作为物理设备,主存储器用随机访问存储器(RAM)构造,这意味着每一段存储器都能够像其他段那样被快速随机访问。主存储器还是易失性的(volatile),无论存储的是什么内容,只要计算机的电源切断,这些内容就都会丢失。当程序正在执行时,该程序以及与之相关的数据总是存储在 RAM 中。计算机 RAM 的大小通常以用户可获得多少 RAM 的字节来标明。当前的 PC 存储器大小至少为 512 兆字节(1 兆字节 = 1 MB)。

第二种类型的存储器为只读存储器(ROM)。ROM 是非易失性的,当电源断开时它的内容不会丢失。因此,ROM 总是包含不能被用户丢失或者改变的基本指令,这些指令包括:电源接通时启动计算机运行所必须的指令,在计算机运行时制造厂商要求必须永久可访问的其他指令。

**中央处理单元** 中央处理单元(CPU)由两个基本子单元组成:控制单元和算术逻辑单元(ALU)。控制单元指导和监督计算机的全部操作。它跟踪为存储器中下一条指令保留的位置、发出从系统的其他单元读取数据和把数据写到系统的其他单元所需要的信号,并且执行所有的指令。ALU 则执行计算机提供的所有计算,如加、减、比较等。

CPU 是计算机的核心元件,是最昂贵的部分。当前的 CPU 利用一片微芯片构造,称为微处理器。图 1.5 展示的是当前笔记本电脑中使用的一个代表技术发展水平的微处理器芯片及其内部结构。

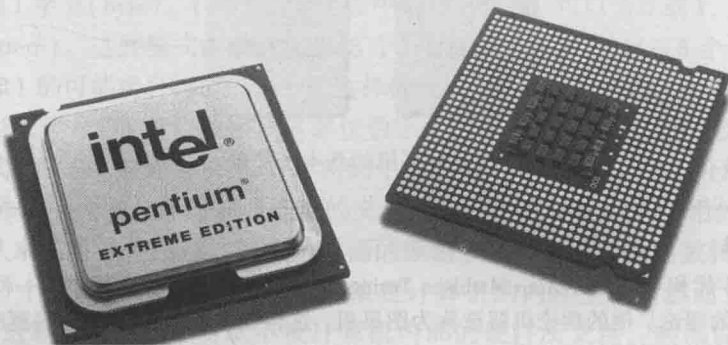


图 1.5 Intel 微处理器展示

**输入/输出单元** 输入/输出(I/O)单元提供对计算机的访问,允许计算机输入和输出数据。这个单元是连接外围设备(如键盘、控制台屏幕和打印机)的接口。