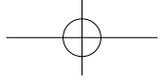


大数据分析基础

李石明 编著

清华大学出版社
北 京



内 容 简 介

本书首先全面介绍了信息技术、计算机基础，以及计算机的起源与发展、计算机系统的组成、操作系统和文件管理等内容，然后深入探讨了Python编程的基础知识，包括编程环境、语法、流程控制、组合数据类型、函数和模块、常用的库等，并通过丰富的实操练习帮助读者掌握Python在文件管理、数据处理、科学计算等领域的应用能力。此外，本书涵盖了机器学习、大数据分析框架及国产大模型DeepSeek等内容，有助于读者建立从基础到前沿的Python知识体系。本书包含丰富的代码示例和综合案例，可以帮助读者快速掌握大数据分析理论和实用的编程技能。

本书适用于高等院校计算机相关专业的学生、Python编程初学者，以及对数据科学、机器学习感兴趣的读者。

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，侵权必究。举报：010-62782989，beiqinquan@tup.tsinghua.edu.cn。

图书在版编目(CIP)数据

大数据分析基础 / 李石明编著. -- 北京：清华大学出版社，2025. 7. -- ISBN 978-7-302-69795-4

I . TP274

中国国家版本馆 CIP 数据核字第 2025ZM5451 号

责任编辑：陈 莉

封面设计：周晓亮

版式设计：方加青

责任校对：成凤进

责任印制：刘海龙

出版发行：清华大学出版社

网 址：<https://www.tup.com.cn>，<https://www.wqxuetang.com>

地 址：北京清华大学学研大厦 A 座 邮 编：100084

社总机：010-83470000 邮 购：010-62786544

投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn

质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn

印 装 者：三河市君旺印务有限公司

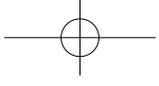
经 销：全国新华书店

开 本：185mm×260mm 印 张：17.25 字 数：388 千字

版 次：2025 年 8 月第 1 版 印 次：2025 年 8 月第 1 次印刷

定 价：59.80 元

产品编号：110831-01



作者简介 AUTHOR

李石明，博士，毕业于中国科学技术大学，主要研究方向为大数据、电子政务、企业管理等学科的教学和研究工作；近年来，主持省部级以上课题4项，参与国家级课题2项，出版《绿色未来：ESG视角下的碳信息披露与企业价值重塑》《电子商务专业办学特色的探索与实践》专著2部，《公司战略与风险管理》《Python编程：从入门到实践》等教材4部，公开发表SCI、CSSCI及北大核心期刊论文12篇。

序言

PREFACE

在数字化浪潮席卷全球的今天，数据已无可争议地成为推动社会进步、经济繁荣与科技创新的核心要素。从微观的企业运营到宏观的国家治理，从日常生活的便捷化到科学研究的深化，数据的力量无处不在，其价值之巨大，堪比工业时代的原油。然而，正如原油须经提炼方能成为动力之源，数据的价值也需要通过深度挖掘、精准分析与有效转化方能显现。这正是大数据分析技术的魅力所在，也是《大数据分析基础》一书的初衷与使命。

在大数据时代的背景下，培养具备跨学科视野、扎实技术基础与创新能力的人才何等重要。Python，这门融合了简洁性与强大功能的编程语言，凭借其丰富的开源生态与高效的开发效率，在大数据处理与分析领域大放异彩，已成为连接理论与实践、学术与产业的桥梁。本书以Python为工具，旨在为读者搭建一个从理论到实践、从基础到进阶的全方位学习平台，助力大家在大数据的海洋中乘风破浪，探索未知。

大数据分析技术的应用，早已超越了单一行业的界限，它正以前所未有的速度渗透到金融、医疗、教育、政务、农业等各个领域，成为推动行业变革与升级的关键力量。然而，面对庞杂的技术栈与多变的应用场景，初学者往往感到无所适从，或是学了一堆技术却不知如何用于解决实际问题。本书正是为了解决这一痛点而著，它摒弃了传统技术书籍单纯罗列知识点的做法，转而以问题为导向，以实战为脉络，通过一系列精心设计的案例，引导读者从真实场景出发，深入理解大数据分析的精髓。

书中，不仅详细介绍了Python编程的基础知识，包括数据友好型语法、Pandas库的使用、Matplotlib与Seaborn的可视化技巧等，还深入探讨了机器学习、分布式计算、大模型构建等前沿话题。通过电商数据分析、社交媒体情感分析、财务大数据分析、政务大数据分析等真实案例，展示了如何将理论知识用于解决实际问题，实现从数据到商业价值的转化。这种“从数据到行动”的闭环思维，不仅能够帮助读者建立扎实的理论基础，更能够培养大家的实践能力和创新思维。

作为教育者，我深知一本好书对于学生成长的重要性。《大数据分析基础》不仅是一本技术书籍，更是一本启发思维、激发潜能的宝典。它鼓励读者先跑通代码，再优化细节；善用工具箱思维，掌握核心范式；从“为什么”到“怎么做”，培养批判性思维；加入社区，保持好奇心，持续精进。通过阅读本书，读者不仅能够掌握大数据分析的核心技能，更能够在实践中不断探索、不断创新，成为推动社会进步的重要力量。

最后，我衷心希望《大数据分析基础》能够成为广大读者探索数据世界的得力助手，助力大家在算法的海洋中锚定方向，在信息的洪流中捕捉价值。携手共进，在这场“解码世界”的旅程中，不断探索、不断前行，共同开创一个更加智慧、更加美好的未来。

黄恒学

2025年5月1日写于北京大学廖凯原楼

前言 PREFACE

在数据驱动的时代，信息技术的每一次革新都在重塑人们的认知范式。从智能手机的普及到人工智能的突破，从物联网到云计算的泛在化，数据已成为推动社会发展的新“原油”。需要指出的是，数据的价值并非与生俱来——它必须经过挖掘、分析和转化，这正是大数据技术的核心使命。

Python作为一门兼具简洁性与强大功能的编程语言，凭借其丰富的开源生态和高效的开发效率，已成为大数据处理与分析领域的“通用语言”。无论是数据清洗、机器学习建模，还是实时流处理、分布式计算，都可以使用Python完成。本书旨在帮助读者跨越理论与实践的鸿沟，掌握使用Python解决实际数据问题的能力。

本书特色

大数据技术的应用早已突破科技公司的边界，融入金融、医疗、教育、政务、农业等各个领域。零售企业需要通过用户行为数据分析优化库存，城市需要通过交通流量数据分析缓解拥堵，医院需要通过患者数据分析预测疾病风险……这些场景的共同点在于：数据是起点，决策才是终点。许多初学者往往陷入两种困境：一是被庞杂的技术栈(如Hadoop、Spark、Flink等)所困扰；二是学了一堆技术，却不知如何解决实际问题。

本书的初衷即在于此——以问题为导向，以实战为脉络。本书摒弃了单纯罗列技术的写法，从真实场景出发，通过实操练习，让读者不仅学会“如何写代码”，而且理解“为什么这样设计”。例如，介绍文本分析时，会从社交媒体评论出发，逐步展示如何用Python提取关键词、分析情感倾向、定位问题根源，最终形成可落地的改进方案。这种“从数据到行动”的闭环思维，正是本书区别于其他相关技术书籍的核心特色。

本书的结构与主要内容

全书围绕基础、工具、应用、拓展四层架构展开，共分为三大部分。

(1) 第1部分“筑基篇——Python与数据科学的桥梁”共包含四章内容，帮助读者奠定知识基础。

第1章从信息技术的演进切入，解析大数据的5V(volume、velocity、variety、veracity、value)特征，并探讨数据驱动决策的底层逻辑。

第2章和第3章深入介绍Python编程，但与传统编程语言书籍不同，本书聚焦“数据友好型”语法，如列表推导式处理多维数据、使用Pandas库实现类SQL操作、使用Matplotlib与Seaborn构建可视化叙事。

第4章揭开机器学习的神秘面纱，通过scikit-learn库的实战，阐释如何用Python训练一个预测模型，并重点讨论过拟合陷阱、特征工程等容易被忽视的实战要点。

(2) 第2部分“进阶篇——分布式计算与生态工具”共包含两章内容，引领读者初步认识工业级大数据处理领域。

第5章深入解析Hadoop与Spark架构设计，避免陷入配置参数细节，通过PySpark实例演示如何用Python调用分布式计算能力。例如，用弹性分布式数据集(RDD)处理TB级日志文件，用MLlib库构建分布式推荐系统。

第6章专章剖析国产大模型DeepSeek，包含从预训练数据集的构建到垂直领域的微调策略等大模型构建过程，并通过医疗影像分析、金融风控等案例，展现国产AI框架的独特优势与技术突破。

(3) 第3部分“实战篇——从数据到商业价值”包含一章内容，是全书的提高篇，介绍了六个案例，均源自真实业务场景。

电商数据分析案例：数据科学中的一个重要应用场景，涉及数据清洗、探索性分析、可视化、用户行为分析、销售趋势分析等。

社交媒体情感分析案例：自然语言处理(NLP)中的一个重要应用场景，旨在通过分析社交媒体上的文本数据(如推文、评论等)来判断用户的情感倾向(正面、负面或中性)。

财务大数据分析案例：数据科学在金融领域的重要应用，涉及财务报表分析、趋势预测、风险评估、投资组合优化等任务。

政务大数据分析案例：利用大数据技术对政府相关数据进行分析，以支持政策制定、资源分配、公共服务优化等决策。

自媒体大数据分析案例：利用大数据技术对自媒体平台(如微博、微信公众号、抖音等)的数据进行分析，以支持内容优化、用户行为分析、趋势预测等任务。

生活服务类大数据分析案例：利用大数据技术对生活服务领域(如餐饮、出行、住宿、娱乐等)的数据进行分析，以支持业务优化、用户行为分析、市场趋势预测等任务。

每个案例均提供完整代码、数据集与商业分析报告模板，读者可一键复现并修改，以适配自身需求。

致读者：如何最大化学习效果

(1) 先跑通，再优化：初次接触代码时，不必纠结于每一行代码的语法，先关注整体流程。本书所有案例均提供最小可行代码(MVP版本)，确保读者快速看到结果，建立正向反馈。

(2) 善用工具箱思维：大数据领域的技术迭代极快，本书强调掌握核心范式，而非死记工具。例如，学完Hadoop MapReduce后，读者应能触类旁通地理解Flink的流处理思想。

(3) 从“为什么”到“怎么做”：第1部分和第2部分中，每章开篇设“灵魂三问”——这项技术解决了什么问题、不用它会怎样、它的局限性在哪里，培养读者的批判性思维。

(4) 加入社区，保持好奇心：技术问题的答案往往不在教科书中。本书鼓励读者参与

GitHub开源项目、关注Kaggle竞赛、订阅权威博客(如Towards Data Science), 在实践中持续精进。

本书免费提供教案、教学大纲、教学课件、习题及解答、源代码, 读者可扫右侧二维码下载。



写在最后

大数据不是冰冷的数字堆砌, 而是人类行为的镜像, 是商业创新的火种, 是社会进步的刻度。学习Python与大数据技术, 本质上是一场“解码世界”的旅程——当你用几行代码从杂乱的数据中提炼出规律, 当你建立的模型帮助一家企业节省了百万元成本, 当你用可视化图表让复杂问题一目了然, 那种创造的喜悦, 正是技术赋予我们的别样浪漫。

在此, 衷心感谢云南大学胡茂老师为本书的撰写和修改提供了大量的宝贵意见和建议。同时, 特别感谢参与本书编写和审阅的清华大学出版社的编辑们, 正是由于他们的辛勤工作和宝贵意见, 本书才得以完善。

希望本书能成为你探索数据世界的指南针, 助你在算法的海洋中锚定方向, 在信息的洪流中捕捉价值。星辰大海, 代码为舟, 让我们共同启航!

李石明
2025年1月

目录 CONTENTS

第1部分 筑基篇——Python 与数据科学的桥梁

第1章 信息技术与计算机基础 / 2

- 1.1 信息社会与计算机 / 3
 - 1.1.1 信息与信息处理 / 3
 - 1.1.2 计算机的起源与发展 / 4
 - 1.1.3 计算机的分类和应用领域 / 9
 - 1.1.4 计算思维 / 11
- 1.2 计算机基础知识 / 13
 - 1.2.1 信息编码 / 13
 - 1.2.2 计算机系统 / 24
 - 1.2.3 操作系统和文件 / 32
- 本章小结 / 37
- 习题 / 38

第2章 Python编程基础 / 39

- 2.1 Python概述 / 40
 - 2.1.1 产生背景 / 40
 - 2.1.2 历史发展 / 40
 - 2.1.3 版本更迭 / 41
 - 2.1.4 应用分析 / 41
 - 2.1.5 发展趋势 / 42
- 2.2 编程环境 / 42
 - 2.2.1 安装Python解释器 / 43
 - 2.2.2 选择合适的IDE / 43
 - 2.2.3 安装常用的库和工具 / 43
 - 2.2.4 配置环境变量 / 43
 - 2.2.5 测试和验证 / 44
- 2.3 基础知识 / 45
 - 2.3.1 标识符 / 45
 - 2.3.2 变量 / 46

- 2.3.3 数据类型 / 46
- 2.3.4 字符串的表示及格式化 / 47
- 2.3.5 语句input/output / 48
- 2.4 流程控制 / 50
 - 2.4.1 程序的基本结构 / 50
 - 2.4.2 分支结构 / 50
 - 2.4.3 循环结构 / 53
 - 2.4.4 综合案例 / 55
- 2.5 组合数据类型 / 61
 - 2.5.1 列表及其操作 / 61
 - 2.5.2 元组及其操作 / 64
 - 2.5.3 字典及其操作 / 66
 - 2.5.4 集合及其操作 / 68
- 2.6 函数和模块 / 71
 - 2.6.1 函数 / 71
 - 2.6.2 实操练习：成绩管理系统 / 77
 - 2.6.3 模块 / 79
 - 2.6.4 实操练习：学生管理系统 / 80
- 2.7 常用的库 / 82
 - 2.7.1 随机数random库 / 82
 - 2.7.2 绘图工具turtle库 / 83
 - 2.7.3 中文分词jieba库 / 85
 - 2.7.4 词云工具wordcloud库 / 85
- 本章小结 / 86
- 习题 / 87

第3章 Python操作实践 / 88

- 3.1 文件和目录操作 / 89
 - 3.1.1 基本概念 / 89
 - 3.1.2 文件的打开与关闭 / 89
 - 3.1.3 文件的读写 / 90
 - 3.1.4 文件和目录的管理 / 91
 - 3.1.5 实操练习：文件和目录管理 / 92
- 3.2 Word文档处理 / 93
 - 3.2.1 安装 python-docx 库 / 93
 - 3.2.2 创建新文档 / 93
 - 3.2.3 修改现有文档 / 94

- 3.2.4 高级功能与扩展功能 / 94
- 3.2.5 实操练习: Word文档处理 / 95
- 3.3 PDF文件处理 / 99
 - 3.3.1 安装 PyPDF2 和 pdfplumber / 99
 - 3.3.2 内容提取 / 99
 - 3.3.3 合并与拆分 / 99
 - 3.3.4 加密与解密 / 100
 - 3.3.5 实操练习: 自动生成报告 / 100
- 3.4 Excel电子表格处理 / 101
 - 3.4.1 安装依赖库 / 101
 - 3.4.2 读取Excel中的数据 / 101
 - 3.4.3 数据清洗与处理 / 101
 - 3.4.4 将数据写入Excel / 101
 - 3.4.5 高级操作 / 101
 - 3.4.6 实操练习: 自动生成销售数据分析报告 / 102
- 3.5 PPT演示文稿处理 / 102
 - 3.5.1 安装 python-pptx 库 / 102
 - 3.5.2 创建基础PPT / 103
 - 3.5.3 高级功能 / 104
- 3.6 NumPy 科学计算库 / 105
 - 3.6.1 NumPy库概述 / 106
 - 3.6.2 数组维度 / 106
 - 3.6.3 数组对象ndarray / 106
 - 3.6.4 数组操作 / 108
 - 3.6.5 数组运算 / 115
 - 3.6.6 数据处理 / 120
 - 3.6.7 实操练习: 酒鬼漫步 / 126
 - 3.6.8 实操练习: 地区经济发展分析 / 127
- 3.7 Pandas 数据处理与分析库 / 129
 - 3.7.1 Pandas库概述 / 130
 - 3.7.2 数据读取与写入 / 130
 - 3.7.3 数据对象DataFrame与Series / 132
 - 3.7.4 实操练习: 将多个DataFrame写入一个Excel的不同sheet的操作 / 144
 - 3.7.5 实操练习: 北京高考分数线统计分析 / 145
 - 3.7.6 数据清洗与预处理(选讲) / 149
 - 3.7.7 实操练习: 预处理销售数据(选讲) / 154
 - 3.7.8 数据聚合与分组(选讲) / 158

- 3.7.9 实操练习：运动员信息的分组与聚合(选讲) / 170
- 3.8 数据可视化工具Matplotlib和Seaborn / 176
 - 3.8.1 Matplotlib基础 / 176
 - 3.8.2 绘制常见图表 / 177
 - 3.8.3 Seaborn高级可视化 / 183
 - 3.8.4 进阶用法 / 190
 - 3.8.5 实操练习：数据可视化 / 192
 - 3.8.6 实操练习：图示分析景点数据 / 193
- 本章小结 / 197
- 习题 / 198

第4章 机器学习基础 / 199

- 4.1 概述 / 200
- 4.2 scikit-learn库 / 201
 - 4.2.1 核心功能 / 201
 - 4.2.2 安装 / 201
 - 4.2.3 实操练习：用KNN算法对鸢尾花数据集进行分类 / 201
- 4.3 监督学习 / 202
- 4.4 无监督学习 / 203
- 4.5 模型评估与优化 / 205
 - 4.5.1 模型评估 / 205
 - 4.5.2 模型优化 / 205
 - 4.5.3 实操练习：用决策树对鸢尾花数据集进行分类 / 206
- 本章小结 / 207
- 习题 / 207

第2部分 进阶篇——分布式计算与生态工具

第5章 大数据基础及应用框架 / 210

- 5.1 大数据基础知识 / 211
 - 5.1.1 概念及特征 / 211
 - 5.1.2 发展现状 / 213
- 5.2 大数据分析理论与方法 / 217
 - 5.2.1 基本理念 / 218
 - 5.2.2 主要步骤 / 219
 - 5.2.3 数据对象 / 220
 - 5.2.4 主要模型 / 221
 - 5.2.5 应用平台 / 228

- 5.3 大数据分析框架 / 229
 - 5.3.1 Hadoop与HDFS / 229
 - 5.3.2 大数据的存储与访问 / 230
- 5.4 Spark数据处理平台 / 230
 - 5.4.1 核心组件 / 231
 - 5.4.2 基本概念 / 231
 - 5.4.3 实操练习：词频统计 / 232
- 5.5 大数据分析应用前沿 / 233
 - 5.5.1 在金融领域的应用 / 233
 - 5.5.2 在零售领域的应用 / 234
 - 5.5.3 在制造业领域的应用 / 235
 - 5.5.4 在医疗领域的应用 / 235
- 本章小结 / 236
- 习题 / 237

第6章 国产大模型DeepSeek / 238

- 6.1 核心特点 / 239
- 6.2 技术基础 / 239
- 6.3 应用场景 / 240
- 6.4 DeepSeek的使用方法 / 240
- 6.5 实操练习 / 240
- 本章小结 / 241
- 习题 / 241

第3部分 实战篇——从数据到商业价值

第7章 实战案例 / 244

- 7.1 案例一：电商数据分析 / 244
 - 7.1.1 示例场景 / 244
 - 7.1.2 安装依赖 / 245
 - 7.1.3 示例代码 / 245
- 7.2 案例二：社交媒体情感分析 / 247
 - 7.2.1 示例场景 / 247
 - 7.2.2 安装依赖 / 247
 - 7.2.3 示例代码 / 247
- 7.3 案例三：财务大数据分析 / 249
 - 7.3.1 示例场景 / 250
 - 7.3.2 安装依赖 / 250

大数据分析基础

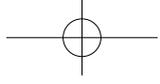
- 7.3.3 示例代码 / 250
- 7.4 案例四：政务大数据分析 / 253
 - 7.4.1 示例场景 / 253
 - 7.4.2 安装依赖 / 253
 - 7.4.3 示例代码 / 253
- 7.5 案例五：自媒体大数据分析 / 255
 - 7.5.1 示例场景 / 256
 - 7.5.2 安装依赖 / 256
 - 7.5.3 示例代码 / 256
- 7.6 案例六：生活服务类大数据分析 / 258
 - 7.6.1 示例场景 / 258
 - 7.6.2 安装依赖 / 259
 - 7.6.3 示例代码 / 259
- 参考文献 / 262



第1部分

筑基篇——Python与数据科学的桥梁





第1章

信息技术与计算机基础

灵魂三问

一问：这项技术解决了什么问题？

信息技术，特别是以计算机为核心的技术，解决了信息的高效收集、处理、存储和传输问题。在信息时代，数据量爆炸性增长，传统的手工处理方式已无法满足需求。计算机和互联网技术的出现，使得人们能够迅速处理和分析海量数据，从中提取有价值的信息，为决策提供支持。同时，信息技术也极大地促进了信息的全球化流通，使得人们能够跨越地域限制，实时获取和分享信息。

二问：不用它会怎样？

如果没有信息技术，我们的工作和生活将受到极大的限制。首先，信息处理效率低下，无法及时响应市场变化和客户需求。其次，信息流通不畅，知识和经验无法有效共享，阻碍了创新和发展。此外，没有信息技术的支持，许多行业(如电子商务、金融科技、远程教育等)将无法存在，人们的生活质量将大幅下降。

三问：它的局限性在哪里？

尽管信息技术带来了诸多便利，但它也存在一些局限性。首先，技术更新迅速，旧的技术和设备容易被淘汰，导致资源浪费和环境污染。其次，信息安全问题日益突出，黑客攻击、数据泄露等事件频发，对个人隐私和企业安全构成威胁。再次，信息技术的发展使得数字鸿沟加大，不同地区、不同群体的信息获取能力差异也在加大。最后，过度依赖信息技术可能导致人类思维能力和创造力的退化，人们可能逐渐失去独立思考和解决问题的能力。

目前，随着互联网技术的快速发展，其应用已深度渗透到社会生活的各个领域。例如，通过便捷的移动终端平台和云端数据服务，人们足不出户即可完成交易支付，获取各类实体商品或虚拟商品；互联网还能基于用户行为分析，提供个性化服务。随着社会对互联网依赖程度的持续加深，网络空间每天都会产生海量数据，对这些数据进行有效分析和处理，能够为各领域提供更精准、更可靠的决策支持与服务。大数据分析、处理及应用正日益成为研究热点，而这一过程离不开以计算机为核心的信息技术。本章将系统介绍信息技术与计算机基础知识，帮助读者构建计算机相关理论框架、理解大数据分析理论与实践背景，为后续章节的学习奠定基础。

1.1 信息社会与计算机

在信息社会中，信息收集、处理和发布需要各种信息技术的支持。信息技术主要包括传感技术、计算机技术、通信技术和控制技术。其中，计算机技术是信息技术的核心。

1.1.1 信息与信息处理

在信息时代，信息的获取极为便捷，然而面对海量且杂乱的数据，人们往往无从下手。如何从庞杂的数据中提取关键信息，将信息转化为知识，进而升华为智慧，成为信息时代的重要课题。

DIKW(data、information、knowledge、wisdom，数据、信息、知识、智慧)体系是人们在探索过程中总结的一种学习方法，如图1-1所示。在信息科学领域，DIKW体系被简称为“信息体系”或“信息金字塔”，清晰地反映了数据、信息、知识、智慧之间的递进关系，使信息更容易被接受、记忆、管理和使用。

数据是DIKW体系中最基础的概念，是信息、知识和智慧的源泉。任何事物的存在方式和运动状态都可以用数据来表示，数据经过加工处理后，具有了知识性并对人类活动产生作用，从而形成信息。

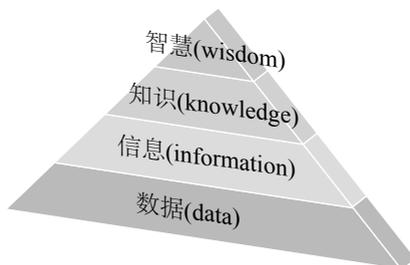


图1-1 DIKW体系

信息可定义为人们对客观事物的属性和运动状态的反映。客观世界中的任何事物都在不停地运动和变化，呈现出不同的状态和特征。可以说信息是有一定含义的、经过加工处理的、对决策有价值的信息，即信息=数据+处理。信息必然来源于数据，经过沉淀且有价值的信息形成知识，用于指导人们的社会活动、经济活动及生产活动。

通常所说的信息处理实际上就是对数据的处理，而它的技术基础是数字电子器件，即利用计算机对各种类型的数据进行处理。

计算机处理信息的过程大体分为数据输入、数据加工和结果输出三个步骤。人们通过输入设备将各种原始数据输入计算机，计算机对输入的数据进行加工处理，然后将结果经由输出设备以文件、图像、动画或声音等形式表示出来。

事实上，计算机与人处理信息的过程有着本质的区别：计算机对信息的处理能力不是

自发产生或学习形成的，而是人事先赋予的。即人设计好程序，再将程序输入计算机，计算机按照程序的规定，一步一步完成程序设计者交给的任务。所以，计算机处理信息的过程就是人所编制的程序的执行过程，是人的思维的一种体现。

计算机是一种能够按照事先存储的程序，自动、高速地进行大量数值计算和信息处理的现代化智能电子设备，处理速度快、计算精度高、存储容量大、逻辑判断能力强、可靠性和通用性强。由于计算机具备类似人脑的记忆和逻辑判断能力，因此也被称为“电脑”。计算机的出现及相关信息技术的快速发展，推动人类社会迅速迈入信息化时代。

1.1.2 计算机的起源与发展

1. 计算机的起源

“计算机”一词的历史可以追溯到1946年，最初定义为“执行计算任务的人”。20世纪40年代以前，为执行计算任务而设计的机器被称为计算器或制表机，而不是计算机。直到第一台电子计算设备问世，人们才开始使用“计算机”这一术语并赋予了其现代定义。

1946年2月，宾夕法尼亚大学莫尔学院电工系和阿伯丁弹道研究实验室历时两年半制造完成世界上第一台电子计算机 ENIAC(electronic numerical integrator and computer, 电子数字积分计算机)，如图1-2所示。该机共使用了18 000个电子管和1800个继电器，每秒运算5000次，每小时耗电150kW，重约30t，占地170m²，长度约30m。研制ENIAC的目的在于计算炮弹及火箭、导弹的弹道轨迹，解决复杂的科学计算问题。这台计算机从1946年2月开始投入使用，到1955年10月最后切断电源，服役9年多。虽然它每秒只能进行5000次加减运算，但它预示了科学家们将从繁重的科学计算工作中解脱出来。ENIAC的问世，标志着电子计算机时代的到来，具有划时代的意义。

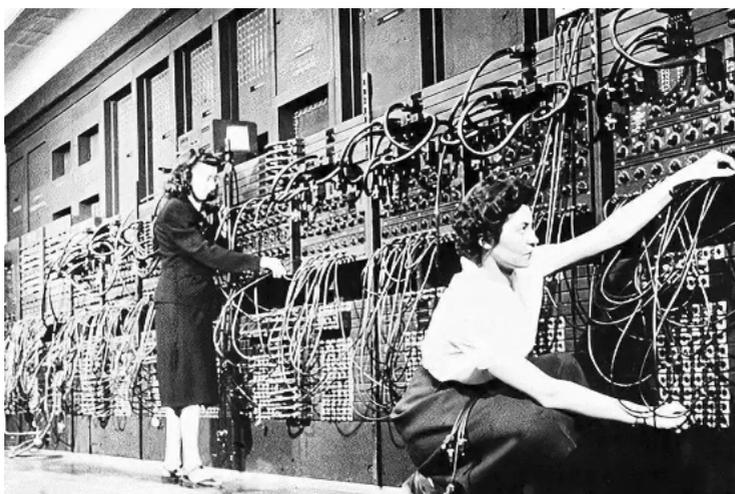


图1-2 世界上第一台电子计算机 ENIAC

ENIAC本身存在两大缺点：一是没有存储器；二是控制非常麻烦，求解问题的程序是通过接线板设定的，问题改变时需要重新接线，有的问题只需要计算几分钟，接线却要花费几小时，计算速度被接线工作抵消了。所以，ENIAC的发明仅仅标志着电子计算机的问世。

研制ENIAC的过程中，美籍匈牙利数学家冯·诺依曼(John von Neumann)总结并提出了两点改进意见：第一，计算机内部直接采用二进制数进行运算；第二，将指令和数据都存储起来，由程序控制计算机自动执行。冯·诺依曼和他的同事们成功研制了第二台电子计算机EDVAC(electronic discrete variable automatic computer，离散变量自动电子计算机)，EDVAC的发明为现代计算机在体系结构和工作原理上奠定了基础，对后来的计算机设计产生了重大影响。EDVAC中采用了“存储程序”的概念，以此为基础的各类计算机统称为冯·诺依曼机。多年来，虽然计算机系统在软硬件各种指标方面与当时的计算机有很大差别，但都属于冯·诺依曼机。不过，冯·诺依曼自己也承认，他的关于计算机“存储程序”的想法来自图灵。

计算机科学的奠基人是英国科学家艾伦·麦席森·图灵(Alan Mathison Turing, 1912—1954)。第二次世界大战期间，图灵设计并完成了真空管机器Colossus，多次成功破译了德军密码，为“二战”的胜利做出了卓越贡献。他在计算机科学领域的主要贡献有两个：一是建立图灵机(Turing machine, TM)模型，奠定了可计算理论的基础；二是提出图灵测试(Turing test)，阐述了机器智能的概念。

图灵机的概念是现代可计算性理论的基础。图灵证明了图灵机能解决的计算问题，实际计算机也能解决；图灵机不能解决的计算问题，实际计算机也无法解决。图灵机的能力概括了数字计算机的计算能力。图灵机对计算机的一般结构、可实现性和局限性都产生了深远的影响。

1950年10月，图灵在哲学期刊*Mind*上发表了一篇著名论文“Computing Machinery and Intelligence”(计算机与智能)。他提出了一个关于判断机器能否思考的实验。图灵认为，如果人与计算机进行文字对话后，人无法判定对方是计算机还是人，那就证明计算机“思考”。今天，人们把这个论断称为图灵测试，它奠定了人工智能的理论基础。

为了纪念图灵对计算机的贡献，美国计算机学会(Association for Computing Machinery, ACM)于1966年创立了“图灵奖”，每年颁发给计算机科学领域的领先研究人员，号称计算机业界和学术界的诺贝尔奖。

2. 计算机的发展

自第一台电子计算机ENIAC诞生以来，计算机以惊人的速度发展。根据计算机所使用的电子元器件不同，计算机的发展经历了传统意义上的四个时代。

1) 第一代：电子管计算机(1946—1957年)

电子管计算机的主要特征是采用电子管作为基本电子元器件，使用机器语言和汇编语言，应用领域主要局限于科学计算。这一代计算机是计算机发展的初级阶段，运算速度每

秒只有几千次至几万次，且体积大、功耗大、价格昂贵、可靠性差，如图1-3所示。

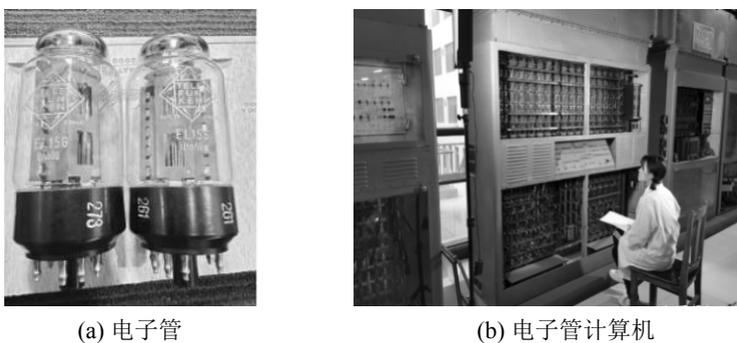


图1-3 电子管和电子管计算机

另外，电子管计算机没有操作系统，由人手工控制作业的输入和输出，通过控制台开关启动程序的运行，如图1-4所示。用户使用电子管计算机的过程大致如下：先把程序纸带装上输入机，启动输入机把程序和数据送入计算机，然后通过控制台开关启动程序，程序计算完毕，用户即可拿走打印结果。

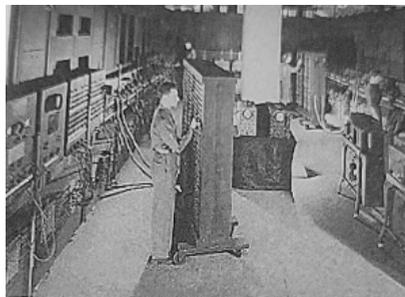


图1-4 操作人员手工控制输入

2) 第二代：晶体管计算机(1958—1964年)

晶体管计算机的主要特征是采用晶体管作为主要元器件，如图1-5所示。这一时期的软件技术出现了程序设计语言(如FORTRAN)和操作系统的雏形(批处理操作系统)。晶体管计算机的主要应用领域由科学计算转为数据处理，与第一代电子管计算机相比，其体积缩小，功耗降低，可靠性有所提高，而运算速度则达到了每秒几万次至几十万次。

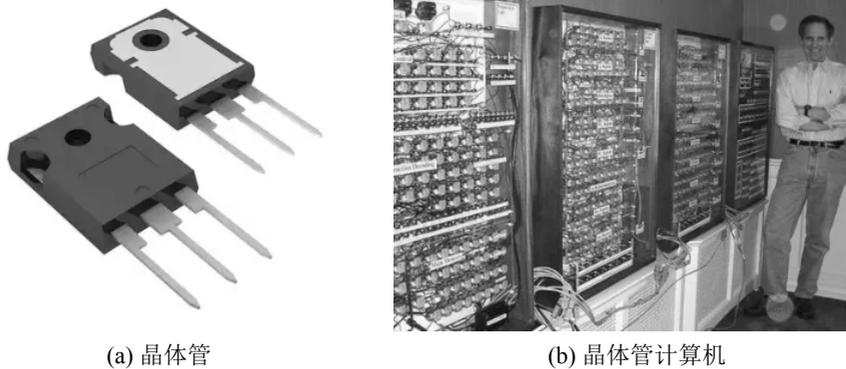


图1-5 晶体管和晶体管计算机

3) 第三代：集成电路计算机(1965—1970年)

集成电路(integrated circuit, IC)产生于1958年，是一种微型电子器件，如图1-6(a)所

示。它的产生揭开了人类电子革命的序幕，同时宣告了数字信息时代的来临。集成电路从1965年开始成为主要元器件。集成电路的发明者是美国工程师杰克·基尔比(Jack Kilby, 1923—2005)，如图1-6(b)所示。他在2000年获得了诺贝尔物理学奖，这是一个迟到了42年的诺贝尔物理学奖。迄今为止，人类的计算机、手机、电视、照相机、DVD及所有电子产品的核心部件都是集成电路。

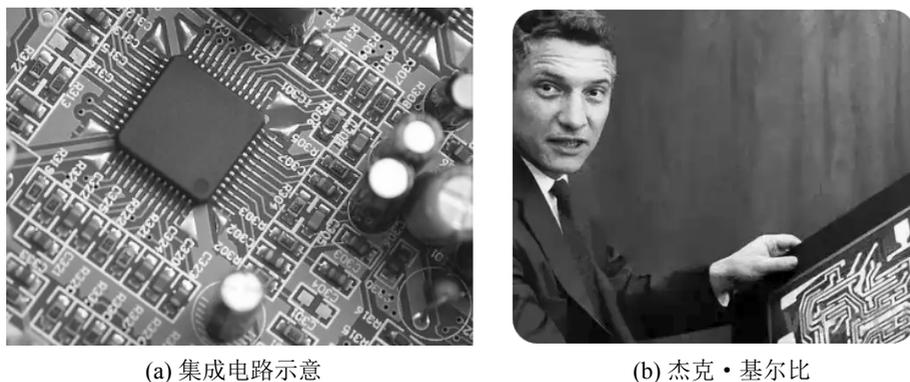


图1-6 集成电路和集成电路的发明者杰克·基尔比

集成电路计算机的主要特征：普遍采用了集成电路，体积、功耗均显著减小，可靠性大大提高；运算速度达到每秒几十万次至几百万次；操作系统的功能日臻完善；出现了多道程序、并行处理技术、虚拟存储系统等。

4) 第四代：大规模或超大规模集成电路计算机(1971年至今)

大规模或超大规模集成电路计算机的主要特征：采用大规模或超大规模集成电路作为计算机的主要元器件；运算速度提高到每秒几百万次至上亿次；随着大规模集成电路技术的发展，微型计算机诞生，它将计算机的运算器与控制器集成在一块芯片上，进一步缩小了计算机的体积并降低了功耗；多机系统和网络化，多处理机系统、分布式系统、计算机网络发展迅速；系统软件的发展不仅实现了计算机运行的自动化，而且推动计算机向工程化和智能化迈进。

3. 未来的计算机

1965年，英特尔(Intel)公司(以下简称英特尔)创始人之一戈登·摩尔(Gordon Moore)提出了被称为计算机第一定律的摩尔定律。该定律指出：集成电路上可容纳的晶体管和电阻的数目将每年增加一倍。1975年，摩尔根据当时的实际情况将该定律进行了修正，把“每年增加一倍”改为“每两年增加一倍”。而普遍被引用的“18个月”的说法，则是由英特尔首席执行官大卫·豪斯(David House)提出的，即预计18个月后芯片的性能将提高一倍(即更多的晶体管使其更快)，是一种成倍数增长的观测。尽管摩尔定律在近现代的数十年间均成立，但它仍应被视为对现象的观测或对未来的推测，是简单评估半导体技术进展的经验法则，而不应被视为一个物理定律或者自然规律。

随着大规模集成电路工艺的发展，芯片的集成度越来越高，但也越来越接近工艺甚

至物理的极限。在传统计算机的基础上，计算机性能的大幅度提高必将遇到难以逾越的障碍。很多专家和学者将目光投向了最基本的物理原理，因为在过去几百年里，物理学原理的应用带来了一系列应用技术的革命，未来以超导计算机、分子计算机、光子计算机和量子计算机为代表的第五代计算机将推动新一轮计算技术的革命。

1) 超导计算机

超导是指在接近绝对零度的温度下，电流在某些介质中传输时所受阻为零的现象。1962年，英国物理学家约瑟夫逊提出了“超导隧道效应”，即对由超导体、绝缘体、超导体组成的器件(约瑟夫逊元件)两端加电压时，电子就会像通过隧道一样无阻挡地从绝缘介质中穿过，形成微小电流，而该器件两端的电压为零。目前制成的超导开关器件，其开关状态切换所需时间已达到0.000 000 000 001s的高水平。这是当今所有电子、半导体、光电器件都无法比拟的，比集成电路要快几百倍。超导计算机运算速度比现在的电子计算机快100倍，而电能消耗仅是电子计算机的1/1000。如果目前一台大中型计算机每小时耗电10kW，那么同样性能的超导计算机只需要一节干电池就可以工作了。

2) 分子计算机

分子计算机就是利用分子计算的能力进行信息的处理。分子计算机的逻辑元件采用生物芯片，生物芯片由生物工程技术产生的蛋白质分子构成。在这种芯片中，信息以波的形式传播，运算速度大大加快，而能量消耗仅相当于普通计算机的1/10，且拥有巨大的存储能力。由于蛋白质分子能够自我组合，再生新的微型电路，使得分子计算机具有生物体的一些特点，能发挥生物体本身的调节机能自动修复芯片故障，能模仿人脑的思考机制。分子计算机是一种由生物分子元件组装成的纳米级计算机，将其植入人体后能自动扫描身体信号、检测生理指标、诊断疾病并控制药物释放等。

3) 光子计算机

光子计算机用光子取代电子进行数据运算、传输和存储，可快速完成复杂的计算工作。在光子计算机中，不同的数据用不同波长的光表示。

与传统的电子计算机相比，光子计算机的优点体现在超高的运算速度、强大的并行处理能力、大存储量、非常强的抗干扰能力、能量消耗小、与人脑相似的容错性等。光子计算机的速度比电子计算机快，光子计算机的计算速度可高达每秒一万亿次，存储量是电子计算机的几万倍，还可以对语言、图形和手势进行识别与合并。

4) 量子计算机

1982年，美国物理学家理查德·费曼(Richard Feynman)提出利用量子体系实现通用计算的新奇想法。随后，英国牛津大学物理学家戴维·多伊奇(David Deutsch)于1985年提出了量子图灵机模型，初步阐述了量子计算机的概念，并提出量子并行处理技术会使量子计算机比传统的计算机功能更强大。

量子计算机是指利用处于多现实态下的量子进行运算的计算机，是一种使用量子逻辑进行通用计算的设备。不同于电子计算机，量子计算用来存储数据的对象是量子比特，具有运行速度快、处理信息能力强和安全性较高等优势。

1.1.3 计算机的分类和应用领域

1. 计算机的分类

计算机是一种用途广泛的机器，有一些特定类型的计算机更适合完成某些特定任务，可按照功能和用途、规模和性能等分成不同的类型。

1) 按照功能和用途划分

(1) 通用计算机。通用计算机的特点是通用性强，具有很强的综合处理能力，能够解决各种类型的问题，既可以进行科学和工程计算，又可以用于数据处理和工业控制等，是一种用途广泛、结构复杂的计算机。

(2) 专用计算机。专用计算机的特点是功能单一，配备解决特定问题的软硬件，能够高速、可靠地解决特定的问题，如数控机床、银行存取款机等。专用计算机针对性强、效率高，结构比通用计算机简单。

2) 按照规模和性能划分

(1) 微型计算机。微型计算机又称个人计算机(personal computer, PC)，是指为满足个人计算需要而设计的一种使用微处理器的计算设备，分为桌面计算机和便携式计算机。

(2) 工作站。工作站既可以指连接到网络的普通个人计算机，也可以指用于高性能计算任务处理的功能强大的桌面计算机，它具有很快的处理速度，能完成医学成像和计算机辅助设计等工作。

(3) 服务器。服务器既可以指计算机硬件，也可以指特定类型的软件，还可以指软件与硬件的结合体，其作用是给网络上的计算机提供数据。任何向服务器请求数据的软件或数字设备都称作客户端。

(4) 大型计算机。大型计算机是指通用性能好、外部设备负载能力强、具有较快的处理速度和较强处理能力的一类计算机，一般作为大型客户机/服务器系统的服务器或者终端/主机系统中的主机。

(5) 巨型机。巨型机也称超级计算机，是指运算速度极快、存储容量大、处理能力极强的计算机，一般用于执行专门的计算密集型任务，如气候研究、密码破译、核武器模拟、石油勘探、天气预报、基因测序等。

(6) 手持计算机。许多手持设备具备计算机大部分的特性，可以接收输入、产生输出、处理数据，并且具有一定的存储能力。不同手持设备的可编程性与多功能性是有差别的。那些允许用户安装应用的手持设备(如智能手机)可以归类为手持计算机。

(7) 可穿戴计算机。近年来，计算机已经小到可以穿戴，并且能配备各种传感器，在医疗保健等场所有许多潜在的用途。

2. 计算机的应用领域

计算机的应用领域从最初诞生时的科学计算扩大到人类社会的各个方面，并改变着人

们传统的工作、学习和生活方式。

1) 科学计算

科学计算也称数值计算，是指计算机用于完成科学研究和工程技术中数学问题的计算。科学计算是研制电子计算机的最初目的，也是计算机最早的应用领域。

2) 数据处理

数据处理是指计算机对大量的数据及时记录、整理、统计并加工成所需要的形式。计算机不仅能用于处理日常的事务，而且支持科学管理与决策，是现代化管理的基础。

3) 过程控制

过程控制也称实时控制，是自动控制原理在生产过程中的应用，现已广泛应用于冶金、石油、化工、水电、纺织、机械、军事和航天等领域。在过程控制中，首先用传感器在现场采集控制对象的数据，求出它们与设定数据的偏差；接着由计算机根据控制模型进行计算，产生相应的控制信号，驱动伺服装置对受控对象进行控制或调整。

4) 计算机辅助系统

计算机辅助系统包括计算机辅助设计(computer aided design, CAD)、计算机辅助制造(computer aided manufacturing, CAM)和计算机辅助教育(computer based education, CBE)等。

5) 人工智能

人工智能(artificial intelligence, AI)是通过计算机模拟人类的智能活动，包括学习、理解、判断、识别、推理和问题求解等。人工智能涉及计算机科学、控制论、信息论、仿生学、神经生理学和心理学等诸多学科，主要应用于机器视觉、指纹识别、人脸识别、视网膜识别、专家系统、自动规划等。近年来，人工智能的研究再次成为热点并取得不少成果，如机器人战胜人类职业围棋选手、无人驾驶汽车广泛应用等。

6) 多媒体技术

多媒体技术是指以数字化为基础，能够对多种媒体信息(包括文字、声音、图形、动画、图像、视频等)进行采集、加工处理、存储和传递，并能使各种媒体信息之间建立起有机的逻辑联系，集成为一个具有良好交互性的系统。

多媒体技术主要涉及数据压缩、多媒体处理(音频信息处理、图像处理)、多媒体数据存储(多媒体数据库)、多媒体数据检索(基于内容的图像检索、视频检索)、多媒体著作工具(多媒体同步、超媒体和超文本)、多媒体通信与分布式多媒体(视频会议系统、视频点播技术等)、多媒体专用设备技术(多媒体专用芯片技术)、多媒体应用技术(远程教学、多媒体远程监控)等。多媒体技术的各方面均涉及一定的规范和标准，根据不同标准制作的多媒体文件，其格式均有所不同。多媒体技术的应用领域主要有知识学习、多媒体出版物、远程医疗、视频会议、语音识别等。

7) 虚拟现实和增强现实

从技术角度讲，虚拟现实(virtual reality, VR)和增强现实(augmented reality, AR)是多媒体技术的发展方向。VR是利用计算机生成的一种模拟环境，通过多种传感设备使用户融入该环境中，实现用户与环境的直接交互，如虚拟课堂、虚拟工厂、虚拟主持人、数字汽

车等。这种模拟环境是用计算机生成的具有表面色彩的立体图形，它可以是现实世界的真实写照，也可以是纯粹构想出来的世界。

在AR系统中，虚拟世界与现实世界叠加在一起，对人们看到的现实世界补充一些有用的信息，例如百度地图的实景路线导航、美图的美颜功能等都用到AR技术。

8) 网络应用

计算机网络是利用通信设备和线路将地理位置不同且功能独立的多个计算机系统互联起来，通过网络软件实现资源共享和信息传递的系统。网络的出现为计算机应用开辟了空前广阔的前景，对人类社会产生了巨大的影响，使人们的生活、工作、学习产生了巨大的变化。

1.1.4 计算思维

计算机原本只是人们解决问题的工具，但当这种工具在几乎每一个领域中都得到广泛使用后，工具就会反过来影响人们的思维方式。2006年，时任美国卡内基-梅隆大学计算机系主任的周以真(Jeannette M. Wing)教授提出了计算思维(computational thinking)的概念，第一次从思维层面阐述了运用计算机科学的基础概念求解问题、设计系统和理解人类行为的过程。

计算思维即抽象实际问题的计算特性，利用计算机求解，涉及如何在计算机中表示问题、如何让计算机通过执行有效的算法来解决问题。计算思维的本质是基于三个阶段的3A迭代过程。

- 抽象(abstraction): 问题的表示。
- 自动化(automation): 解决方案的表达。
- 分析(analyse): 解决方案的执行和评估。

从问题的计算机表示、算法设计直到编程实现，计算思维贯穿全过程。

1. 常见思想和方法

基于计算机的能力和局限性，计算机科学家提出了很多关于计算的思想和方法，从而开发了一整套利用计算机解决问题的思维工具。下面简要介绍不同阶段的常见思想和方法。

1) 问题表示

用计算机解决问题，首先要建立问题的计算机表示。抽象是用于问题表示的重要思维工具。例如，小学生经过学习都知道应用题“原来有五个苹果，吃掉两个后还剩几个”可以抽象表示成“5-2”，这里显然只抽取了问题中的数量特性，完全忽略了苹果的颜色或吃法等相关特性。一般意义上的抽象是指忽略研究对象的具体的或无关的特性，而抽取其一般的或相关的特性。计算机科学中的抽象包括数据抽象和控制抽象，简言之就是将现实世界中的各种数量关系、空间关系、逻辑关系和处理过程等表示成计算机世界中的数据

结构(数值、字符串、列表、堆栈、树等)和控制结构(基本指令、顺序执行、分支、循环、模块等),即建立实际问题的计算模型。另外,抽象还用于在不改变意义的前提下隐去或减少过多的具体细节,以便每次只关注少数几个特性,从而有利于理解和处理复杂系统。显然,通过抽象还能发现不同问题的共性,从而建立相同的计算模型。总之,抽象是计算机科学中广泛使用的思维方式。

可以在不同层次上对数据和控制进行抽象,不同抽象级对问题进行不同颗粒度或详细程度的描述。人们经常在较低抽象级之上再建立一个较高的抽象级,以便隐藏较低抽象级的复杂细节,提供更简单的求解方法。例如,在互联网上发送一封电子邮件实际上要经过不同抽象级的多层网络协议才得以实现,写邮件的人肯定不希望先掌握网络底层知识才能发送邮件。再如,人们经常在现有软件系统之上搭建新的软件层,目的是隐藏底层系统的观点或功能,提供更便于理解或使用的新观点或新功能。

2) 算法设计

问题得到表示之后,接下来的关键是找到问题的解法——算法。算法设计是计算思维大显身手的领域,计算机科学家采用多种思维方式和方法来实现有效的算法。例如,利用分治法的思想找到了高效的排序算法,利用递归思想轻松地解决了Hanoi塔问题,利用贪心法寻求复杂路网中的最短路径,利用动态规划方法构造决策树,等等。计算机在各个领域的成功应用都有赖于高效算法的发现,而高效算法又依赖于各种算法设计方法的巧妙运用。

3) 编程实现

找到了解决问题的算法,接下来就要用编程语言来实现算法,这个领域同样是各种思想和方法的宝库。例如,类型化与类型检查方法将待处理的数据划分为不同的数据类型,编译器或解释器可以借此发现很多编程错误,这和自然科学中量纲分析的思想是一致的;又如,结构化编程方法使用规范的控制流程来组织程序的处理步骤,形成层次清晰、边界分明的结构化构造,每个构造具有单一的入口和出口,从而使程序易于理解、排错、维护和验证正确性;再如,模块化编程采取从全局到局部的自顶向下设计方法,将复杂程序分解成许多较小的模块,完成了所有底层模块后,将模块组装起来即构成最终程序;再如,面向对象编程以数据和操作融为一体的对象为基本单位来描述复杂系统,通过对象之间的协作和交互实现系统的功能。本书后续章节中所用的Python即支持面向对象的高级程序设计语言。

4) 可计算性与算法的复杂性

用计算机解决问题时,不仅要找到正确的解法,还要考虑解法的复杂度。这和数学思维不同,因为数学家可以满足于找到正确的解法,但不会因为该解法过于复杂而抛弃不用。但对计算机来说,如果一个解法太复杂,导致计算机要耗费几年、几十年乃至更久才能得出结果,那么这种“解法”只能被抛弃,问题等于没有解决。有时即使一个问题已经有了可行的算法,计算机科学家仍然会寻求更有效的算法。

虽然很多问题对计算机来说难度太高甚至是不可能完成的任务,但计算思维具有灵活、变通、实用的特点,对这样的问题可以寻求不那么严谨但现实可行的实用解法。例如,当计算机有限的内存无法容纳复杂问题中的海量数据时,计算机科学家设计出了缓冲

方法来分批处理数据。当许多用户共享并竞争某些系统资源时，计算机科学家又利用同步、并发控制等技术来避免竞态和僵局。

2. 日常生活中的计算思维

人们在日常生活中的很多思维方式其实都和计算思维不谋而合，也可以说计算思维从生活中吸收了很多有用的思想和方法，部分例子如下。

算法过程：菜谱可以说是算法(或程序)的典型代表，它将一道菜的烹饪步骤一步一步地罗列出来，即使不是专业厨师，照着烹饪步骤也能做出可口的菜肴。这里，菜谱的每一步骤都必须足够简单、可行。例如，“将土豆切成块状”“将一两油入锅加热”等都是可行的步骤，而“使菜肴具有神秘香味”则不是可行的步骤。

查找：如果要在英汉词典中查一个英文单词，相信没有人会从第一页开始逐页翻看，而会根据字典是有序排列的事实，快速地定位单词词条。又如，如果老师说“请翻到本书第4章”，则可以通过书前的目录直接找到第4章所在的页码。这正是计算机中广泛使用的索引技术。

回溯：人们在路上遗失了东西之后，会沿原路边往回走边寻找。或者，人们要前往某目的地，到达一个岔路口后，会选择一条路走下去，如果发现此路不通就会原路返回，回到岔路口后选择另一条路。这种回溯法对于搜索问题是非常重要的。

缓冲：学生随身携带所有的教科书是不可能的，因此每天只能把当天要用的教科书放入书包，第二天再换入新的教科书，这就是缓冲。

并发：烧菜时，如果一个菜需要在锅中煮一段时间，厨师会利用这段时间去做别的事情(例如将另一个菜洗净、切好)，不会无所事事地等待。在此期间，如果锅里的菜需要加盐和佐料，厨师可以放下手头的活儿去处理锅里的菜。虽然只有一个厨师，但他可以同时做几个菜。

类似的例子还有很多。要强调的一点是，学习用计算机解决问题时，可以参考生活中遇到类似问题时的做法，一定会对找出问题解决方法有所帮助。

1.2 计算机基础知识

1.2.1 信息编码

1. 数字化

计算机和其他数字设备一样能处理文本、图像、语音和视频等各种信息，而这些信息最终都转换为简单的电脉冲，并以0和1序列的形式存储起来。将信息用0和1这两个符号构成的符号串来表示的过程称为编码。对数据编码后进行处理、存储、传递称为信息的数字化。

数字化的一个显著优势就是，书籍、电影、歌曲、通话、文档和照片等各种不同的内容都可以转换为同一类信号，这些信号不需要单独的设备来处理。数字化技术出现之前，电话通话需要电话机和专门的电话线路，浏览照片需要幻灯片投影仪和投影幕布，阅读需要纸质书籍，拍照需要相机和胶卷，看电影则需要胶片放映机。数字化技术出现后，通话、照片、书籍和电影都可以由一个设备来管理，并可以通过一组通信线路来传输。

数据的类型有很多，数字和文字是最简单的类型，表格、声音、图形和图像则是复杂的类型，因此编码是一件非常重要的工作，要考虑数据的特性并便于计算机的存储和处理。

2. 进制及其转换

1) 数的进制

计算机中存放的是二进制数，为了方便书写和表示，还引入了八进制数和十六进制数。无论哪种数制，其共同之处都是进位计数制。

一般来说，如果数制只采用 R 个基本符号(如 $0, 1, 2, \dots, R-1$)表示数值，则称数值为“ R 进制数”， R 为该数值的“基数”。例如，十进制数采用10个基本符号($0, 1, \dots, 9$)，其基数为10；二进制数采用2个基本符号($0, 1$)，其基数为2。数值中每一个固定位置对应的单位值称为“权”。

【例1-1】将十进制数 $(368.19)_{10}$ 按权展开。

$$368.19 = 3 \times 10^2 + 6 \times 10^1 + 8 \times 10^0 + 1 \times 10^{-1} + 9 \times 10^{-2}$$

任意一个 R 进制数 N 可表示为

$$N = a_{n-1} \times R^{n-1} + a_{n-2} \times R^{n-2} + \dots + a_1 \times R^1 + a_0 \times R^0 + a_{-1} \times R^{-1} + \dots + a_{-m} \times R^{-m} = \sum_{i=-m}^{n-1} a_i \times R^i$$

其中， a 是数码， R 是基数， R^i 是权； m 和 n 为正整数， n 为小数点左边的位数， m 为小数点右边的位数。

不同的基数表示不同的进制数。例如：

$$(123.45)_8 = 1 \times 8^2 + 2 \times 8^1 + 3 \times 8^0 + 4 \times 8^{-1} + 5 \times 8^{-2}$$

其中，下标 O 表示该数是八进制数。

通常用下标 B (或2)、 O (或8)、 D (或10)、 H (或16)表示该数是二进制数、八进制数、十进制数和十六进制数。

计算机中常用的各种进制数的表示如表1-1所示。

表1-1 计算机中常用的各种进制数的表示

说明	二进制数	八进制数	十进制数	十六进制数
规则	逢二进一	逢八进一	逢十进一	逢十六进一
基数	2	8	10	16
基本符号	0, 1	0, 1, 2, ..., 7	0, 1, 2, ..., 9	0, 1, 2, 9, A, B, ..., F
权	2^i	8^i	10^i	16^i
表示形式	下标 B (或2)	下标 O (或8)	下标 D (或10)	下标 H (或16)

2) R 进制数转换为十进制数

基数为 R 的数值，只要将各位数码与它的权相乘，其积相加，和数就是十进制数。展开式为

$$N = \sum_{i=-m}^{n-1} a_i \times R^i$$

【例1-2】将二进制数 $(10001100.101)_B$ 转换为十进制数。

$$\begin{aligned} (10001100.101)_B &= 1 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\ &= 128 + 0 + 0 + 0 + 8 + 4 + 0 + 0 + 0.5 + 0 + 0.125 \\ &= 140.625 \end{aligned}$$

因此， $(10001100.101)_B = (140.625)_D$ 。

【例1-3】将八进制数 $(167)_O$ 转换为十进制数。

$$(167)_O = 1 \times 8^2 + 6 \times 8^1 + 7 \times 8^0 = (119)_D$$

【例1-4】将十六进制数 $(3A7)_H$ 转换为十进制数。

$$(3A7)_H = 3 \times 16^2 + A \times 16^1 + 7 \times 16^0 = (935)_D$$

3) 十进制数转换为 R 进制数

将十进制数转换为 R 进制数时，可将此数分成整数与小数两部分分别转换，然后拼接而成。十进制数整数部分转换为 R 进制数整数，采用除 R 取余法：用十进制整数连续地除以 R 取余数，直到商为0，余数从右到左排列，第一次取得的余数为最低位，最后所得余数为最高位；小数部分转换为 R 进制数采用乘 R 取整法：将十进制小数不断乘以 R 取整数，直到小数部分为0或达到所要求的精度为止，所得的整数在小数点后自左向右排列。

【例1-5】将 $(123.125)_D$ 转换为二进制数。

取余数		取整数	
2 123	1	0.125	
2 61	1	×2	
2 30	0	0.250	0
2 15	1	×2	
2 7	1	0.500	0
2 3	1	×2	
2 1	1	1.000	1
0			

$$(123.125)_D = (1111011.001)_B$$

类似地，将十进制数234.12转换为八进制数的结果为

$$(234.12)_D = (352.075)_O$$

4) 二进制数、八进制数、十六进制数之间的转换

由前面的例子可以看到，将十进制数转换为二进制数，转换的书写过程较长。二进制数比等值的十进制数占更多的位数，容易出错。为方便起见，可借助八进制数和十六进制数进行转换或表示。由于二进制数、八进制数和十六进制数间存在特殊关系： $2^3=8$ ， $2^4=16$ ，即1位八进制数相当于3位二进制数，1位十六进制数相当于4位二进制数，因此转

换时就容易。它们之间的关系见四种进制数对照表，如表1-2所示。

表1-2 四种进制数对照表

十进制数	二进制数	八进制数	十六进制数	十进制数	二进制数	八进制数	十六进制数
0	0000	00	0	8	1000	10	8
1	0001	01	1	9	1001	11	9
2	0010	02	2	10	1010	12	A
3	0011	03	3	11	1011	13	B
4	0100	04	4	12	1100	14	C
5	0101	05	5	13	1101	15	D
6	0110	06	6	14	1110	16	E
7	0111	07	7	15	1111	17	F

根据这种对应关系，二进制数转换为八进制数时，以小数点为中心分组，每3位为一组，两头不足3位则补0即可。同样，二进制数转换为十六进制数以每4位为一组，两头不足4位则补0即可。

【例1-6】将二进制数 $(1011111011.0011001)_B$ 转换为十六进制数(见表1-3)。

表1-3 二进制数转换为十六进制数

0010	1111	1011	0011	0010	二进制数
↓	↓	↓	↓	↓	↓
2	F	B	3	2	十六进制数

因此， $(1011111011.0011001)_B=(2FB.32)_H$ 。

【例1-7】将二进制数 $(1011111011.0011001)_B$ 转换为八进制数(见表1-4)。

表1-4 二进制数转换为八进制数

001	011	111	011	001	100	100	二进制数
↓	↓	↓	↓	↓	↓	↓	↓
1	3	7	3	1	4	4	八进制数

因此， $(1011111011.0011001)_B=(1373.144)_O$ 。

同样，将八(十六)进制数转换为二进制数只要将1位转换为3(4)位即可，中间的0不能省略，小数点前的高位0和小数点后的低位0可以去掉。

【例1-8】将十六进制数 $(1A3D.B2)_H$ 转换为二进制数(见表1-5)。

表1-5 十六进制数转换为二进制数

1	A	3	D	B	2	十六进制数
↓	↓	↓	↓	↓	↓	↓
0001	1010	0011	1101	1011	0010	二进制数

因此, $(1A3D.B2)_H=(1101000111101.1011001)_B$ 。

3. 数据信息编码

1) 计算机中数据的存储单位: 位、字节与字长

(1) 位(bit, b)。位是计算机中表示信息的最小单位, 代码为0和1。 n 位二进制数能表示 2^n 种状态。

(2) 字节(byte, B)。字节是计算机中存储信息的基本单位, 每字节由8位二进制数组成。计算机是以字节为单位计算存储容量的。一个英文字母(不区分大小写)占1字节的空
间, 一个中文汉字占2字节的空
间。英文标点占1字节, 中文标点占2字节。

换算关系如下:

$$\begin{aligned} 1\text{B} &= 8\text{b} \\ 1\text{KB} &= 1024\text{B} = 2^{10}\text{B} \\ 1\text{MB} &= 1024\text{KB} = 2^{20}\text{B} \\ 1\text{GB} &= 1024\text{MB} = 2^{30}\text{B} \\ 1\text{TB} &= 1024\text{GB} = 2^{40}\text{B} \\ 1\text{PB} &= 1024\text{TB} = 2^{50}\text{B} \\ 1\text{EB} &= 1024\text{PB} = 2^{60}\text{B} \end{aligned}$$

(3) 字长。计算机进行数据处理和运算的单位, 即CPU在单位时间内能一次处理的二进制数据的位数, 称为字长。字长由若干字节组成, 如16位、32位、64位等。目前常用的是32位计算机和64位计算机。字长较长的计算机在相同的时间内能处理更多的数据。字长是衡量计算机性能的重要指标。

2) 机器数

前面提到, 计算机内部采用二进制表示各类数据。对于数值型数据, 数据有正负和小数之分, 因此, 必须解决符号位、小数点在计算机内部的表示问题。

通常, 把一个数在计算机内的二进制表示形式称为机器数, 该数称为这个机器数的真值。一个机器数一般由三类符号构成: 数字0和1(表示符号位的+和-), 以及小数点。数字0和1的二进制编码是直接的, 剩下的就是解决符号位和小数点的表示问题。

机器数具有如下三个特点。

(1) 由于计算机设备的限制和操作上的便利, 机器数有固定的位数。

机器数所表示的数受到位数的限制, 在一定的范围内, 超过这个范围就会产生“溢出”。例如, 一个8位的机器数, 所能表示的无符号整数的最大值是全1(11111111), 即十进制数255。如果超过这个值, 就会产生“溢出”。

(2) 机器数把其真值的符号数字化。

通常机器数中规定的符号位(一般是最高位)取0或1, 分别表示其值的正或负(0表示正数, 1表示负数)。例如, 一个8位机器数, 其最高位是符号位, 对于00101110和10010011, 其真值分别为十进制数+46和-19。

(3) 机器数中，采用定点和浮点方式来表示小数点的位置。

定点表示法是将小数点的位置固定在一个二进制数的某一位置。定点数分为定点纯小数(小数点固定在符号位之后，数的最前面)和定点整数(小数点固定在数据最后一位之后，表示一个纯整数)。

浮点表示法是指表示一个数时，其小数点的位置是浮动(可变)的，是数的科学(指数)计数法在计算机中的具体实现。浮点表示法表示数的范围较大，但运算规则复杂，运算速度相对来说较慢。

3) 原码、反码与补码

(1) 原码。带符号的机器数称为数的原码。实际上，计算机中不是用原码存储有符号数的。为什么呢？机器数在进行运算时，若将符号位和数值位同时参与运算，则会得出错误的结果。例如：

$$\begin{aligned} X &= +6 & [X]_{\text{原}} &= 00000110 \\ Y &= -3 & [Y]_{\text{原}} &= 10000011 \\ X+Y &= +6+(-3) & &= 6-3=3 \end{aligned}$$

原码相加，得到-9：

$$\begin{array}{r} 00000110 \\ + 10000011 \\ \hline 10001001 \cdots \cdots -9 \end{array}$$

原码相减，得到-3：

$$\begin{array}{r} 00000110 \\ - 10000011 \\ \hline 10000011 \cdots \cdots -3 \end{array}$$

因此，为了运算方便，计算机中引入了反码和补码的概念，将加减法运算统一转换为补码的加法运算。

正数的原码、反码和补码形式完全相同，负数则有不同的表示形式。

整数X的原码表示：整数的符号位用“0”表示正，“1”表示负，其数值部分是该数的绝对值的二进制表示。

$$[X]_{\text{原}} = \begin{cases} 0X & X \geq 0 & +7: 00000111 & +0: 00000000 \\ 1|X| & X \leq 0 & -7: 10000111 & -0: 10000000 \end{cases}$$

表示数的范围为-127~127(11111111~01111111)，在原码表示中，0有两种表示方法。

(2) 反码。反码是求补码的中间过渡。负数的反码是对该数的原码中除符号位外的各位取反。

$$[X]_{\text{反}} = \begin{cases} 0X & X \geq 0 & +7: 00000111 & +0: 00000000 \\ 1|X| & X \leq 0 & -7: 11111000 & -0: 10000000 \end{cases}$$

在反码表示中，0有两种表示方法。

(3) 补码。负数的补码是在其反码的基础上，在末位加1。

$$[X]_{\text{补}} = \begin{cases} 0X & X \geq 0 & +7: 0000111 & +0: 0000000 \\ 1|X|+1 & X \leq 0 & -7: 1111001 & -0: 0000000 \end{cases}$$

补码表示中，0有唯一的表示形式，即 $[+0]=[-0]=00000000$ ，因此，可以用多出来的编码10000000来扩展补码的表示范围，最高位1既可看作符号位负数，又可表示数值。所以对于一个八位的二进制机器数，补码表示数的范围为-128~127。这就是补码与原码、反码最小值不同的原因。

【例1-9】 利用补码进行(+6)+(-6)运算。

$$X=+6 \quad [X]_{\text{原}}=00000110 \quad [X]_{\text{补}}=00000110$$

$$Y=-6 \quad [Y]_{\text{原}}=10000110 \quad [Y]_{\text{补}}=1111010$$

两数相加：

$$\begin{array}{r} 00000110 \cdots \cdots +6 \text{的补码} \\ + 1111010 \cdots \cdots -6 \text{的补码} \\ \hline 00000000 \cdots \cdots 0 \text{的补码} \end{array}$$

【例1-10】 利用补码进行(+6)+(-3)运算。

$$X=+6 \quad [X]_{\text{原}}=00000110 \quad [X]_{\text{补}}=00000110$$

$$Y=-3 \quad [Y]_{\text{原}}=10000011 \quad [Y]_{\text{补}}=1111101$$

两数相加：

$$\begin{array}{r} 00000110 \cdots \cdots +6 \text{的补码} \\ + 1111101 \cdots \cdots -3 \text{的补码} \\ \hline 00000011 \cdots \cdots +3 \text{的补码} \end{array}$$

数的原码、反码和补码表示总结如下。

- 对于正数，其原码、补码和反码的表示相同，即 $[X]_{\text{原}}=[X]_{\text{补}}=[X]_{\text{反}}$ ，符号位用0表示，其余各位为该数的绝对值。
- 对于负数：①原码，符号位用“1”表示，加上该数的绝对值；②反码，符号位“1”不变，其余各位求反；③补码， $[X]_{\text{补}}=[X]_{\text{反}}+1$ (末位加1)。

4. 字符编码

1) ASCII码

ASCII码，即 American Standard Code for Information Interchange(美国信息交换标准代码)，是西文字符的一种编码规范，原为美国国家标准，1967年被国际标准化组织(ISO)定为国际标准。ASCII码是1字节编码，编码范围是0~255，最多可表示256个不同字符。具有256个编码的ASCII码分为两大部分：标准ASCII码和扩充ASCII码。

(1) 标准ASCII码。在ASCII码中，二进制最高位为0的编码称为标准ASCII码，其编码范围是十进制0~127，即标准ASCII码有128个编码。可见，标准ASCII码只需要7位二进制

字符进行编码就可以了，所以又称为7位字符编码。而在实际存储时，由于存储器是按字节作为最小单位来组织的，7位编码仍然需要占1字节的存储空间，必须在编码前补一个二进制0成为1字节。标准ASCII码如表1-6所示。

表1-6 标准ASCII码

低四位	高四位							
	0000	0001	0010	0011	0100	0101	0110	0111
0000	NULL	DLE	空格	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	“	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BELL	ETB	'	7	G	W	g	w
1000	BS	CAN	(8	H	X	h	x
1001	HT	EM)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

这样，英文中的每一个字符都有一个固定的编码，保存字符时只需要保存它的ASCII码即可。

ASCII码表中有33个控制符编码(00H~1FH、7FH)和95个可显字符编码(20H~7EH)。它确定了西文字符的顺序：小写字母在大写字母之后，其顺序与字母顺序一致。不难发现，只要记住字母“A”“a”和数字“0”的ASCII码，就能够推算出所有英文大、小写字母和数字的ASCII码。

(2) 扩充ASCII码。扩充ASCII码的二进制最高位为1，其范围为128~255。扩充的ASCII码也是128个，虽然这些代码也有国际标准，但它们是可变字符。各国都利用扩充ASCII码来定义自己国家的文字代码。例如，日本将其定义为片假名字符，我国则将其定义为中文文字的代码。

2) 汉字编码

ASCII码只对英文字母、数字和标点符号等进行了编码。为了用计算机处理汉字，同样需要对汉字进行编码。由于汉字是象形文字，种类繁多，远比西文复杂，而且在一个汉字处理系统中，输入、内部处理、输出对汉字编码的要求不尽相同，因此要进行一系列的汉字编码及转换，即必须解决汉字的输入码、交换码、机内码和字形码等问题。

(1) 输入码。为了使用计算机的西文键盘输入汉字，必须提供汉字的输入编码，即汉

字的输入码，也称外码。一般来说，汉字输入码应具有单一性、方便性、高速性和可靠性。目前，有多种汉字输入编码。

① 数字编码。数字编码是用等长的数字串为汉字逐一编号，以这个编号作为汉字的输入码。例如，区位码、电报码等都属于数字编码。数字编码规则简单，易于和汉字的内部码互相转换，但难记忆，不宜推广使用。

1980年，为了使每个汉字有一个全国统一的代码，我国发布了《信息交换用汉字编码字符集 基本集》，即国家标准GB 2312—1980，它是我国国家标准简体中文字符集。新加坡等国家和地区也采用此编码，是目前使用最多的汉字编码标准。

该标准基于区位码设计，将编码表分为94个区，每个区对应94个位，每个位放一个字符(包括汉字、符号、数字等)。这样每个字符的区号和位号组合起来就成为该汉字的区位码。区位码一般用十进制来表示。例如，“啊”字位于16区01位，它的区位码就是1601。为了处理与存储的方便，每个汉字的区号和位号在计算机内部分别用一个字节来表示。

在GB 2312—1980中，01~09区是符号区、数字区，16~87区是汉字区，10~15和88~94是未定义的空白区。其中，共有6763个简化汉字(分为两级，第一级3755个汉字，属于常用汉字，按汉字拼音字母顺序排列；第二级3008个汉字，属于次常用汉字，按部首排列)和682个汉字符号。

② 字音编码。字音编码是以汉字读音为基础的一种编码，常用的是拼音码。拼音码简单、易学，用户只需要能正确写出汉字的拼音即可使用。由于同音汉字较多，拼音码的重码率较高，输入时常要进行屏幕选字，对汉字输入速度有影响。

③ 字形编码。字形编码用字母表示组成汉字的基本笔画，按汉字基本笔画的书写顺序和组成进行编码，如五笔字型码、表形码等。它的特点是输入速度较快，重码率低，但是由于要对汉字进行拆分，因此需要学习如何拆字、记忆字根等。

需要指出的是，以上所说的均是编码输入的方式，目前还有一些非编码输入方式，如手写板输入、光学字符识别输入、语音输入等，均是模式识别输入方式，这些方法大大方便了汉字的输入操作。与编码输入的确定性不同，模式识别输入存在不确定性，识别难度较大，对算法要求较高。

(2) 交换码。交换码是指具有汉字处理功能的不同计算机系统之间交换汉字信息时所使用的编码标准。我国一直沿用GB 2312—1980所规定的国标码作为统一的汉字信息交换码。

由于区位码无法用于汉字通信，因为它可能与通信使用的控制码(ASCII码表中的控制字符00H~1FH，即0~31)发生冲突，于是，ISO 2022规定每个汉字的区号和位号必须分别加上20H(即十进制的32)，得到对应的国标交换码，简称国标码，也称交换码。国标码通常用十六进制来表示。例如，“啊”字的区位码(16 01)的十六进制表示为1001H，其国标码为3021H。

(3) 机内码。由于文本中通常混合有汉字和西文字符，汉字信息如果不加以特别标识，就会与单字的ASCII码混淆。例如，“啊”字的国标码为3021H，用来表示其国标码的两个

字节最高位均为0，即两个字节分别为30H和21H，可以解析成两个字符“0”和“!”。

为了解决这一问题，将一个汉字看成两个扩展ASCII码，将GB 2312—1980中汉字的两个字节的最高位都置1，即国标码加上80H(即二进制数10000000，十进制的128)。这种高位为1的双字节汉字编码即GB 2312—1980汉字的机内码，简称内码，又称汉字ASCII码。它是计算机内部存储、处理加工和传输汉字时所用的代码。每个汉字的外码可以有多种，但是内码只有一个。

【例1-11】“啊”字的区位码(16 01)的十六进制表示为1001H，国标码为3021H，机内码则为B0A1H(30H+80H=B0H，21H+80H=A1H)。

区位码是一种早期的汉字输入编码，由区码(十进制)和位码组成，例如1601表示第16区第1位的汉字。使用时需要先将区码和位码分别转换为十六进制(如1601→1001H)，这是后续转换的基础。国标码通过区位码转换得到：将十六进制区位码的区号和位号分别加上2020H(即整体加2020H)，例如1001H + 2020H = 3021H。机内码是计算机内部处理汉字的实际编码，由国标码转换而来：国标码每个字节最高位置1(即加8080H)，例如3021H + 8080H = B0A1H。

汉字机内码、国标码和区位码三者之间的关系为

$$\text{国标码} = \text{区位码} + 2020\text{H}$$

$$\text{机内码} = \text{国标码} + 8080\text{H}$$

$$\text{机内码} = \text{区位码} + \text{A0A0H}$$

【例1-12】求汉字“爪”的机内码，可以先对照GB 2312—1980，找到该字的区位码5506，对应的十六进制表示为3706H，则其机内码为3706H+A0A0H=D7A6H。

除了GB 2312—1980编码，常用的汉字编码标准还有如下几个。

Big5：又称大五码或五大码，是繁体中文社区中最常用的汉字字符集标准，共收录13 053个汉字，使用2字节表示。

GBK：为了更好地满足古籍研究等方面的文字处理需要，我国在1995年颁布了GBK汉字内码扩充规范，不仅包含GB 2312—1980中规定的全部汉字和符号，还收录了包括繁体字在内的大量汉字和符号。它是GB 2312—1980的扩展，共收录了21 003个文字，支持国际标准ISO 10646中的全部中、日、韩文字，也包括Big5编码中的所有汉字，使用2字节编码。

GB18030：对GBK的扩充，覆盖中国少数民族文字和繁体汉字。其编码空间约为161万码位，收录了70 244个文字。它采用变长多字节编码，每个字可以由1字节、2字节或4字节组成。

当采用GB 2312—1980、GBK和GB 18030三种不同的汉字编码标准时，一些常用的汉字如“中”“国”等，它们在计算机中的表示(内码)是相同的。

(4) 字形码。字形码是为了解决汉字的显示和打印等输出问题而制定的编码标准。汉字的字形码表示汉字字形的字模数据，又称汉字字模。字模通常有点阵和矢量两种汉字字形码。点阵码主要用于显示输出，其他类型的输出主要使用点阵码和矢量码。

用点阵表示字形时，汉字字形码指的就是这个汉字字形点阵的代码，如图1-7所示。

系统提供的所有汉字字形码的集合组成了系统的汉字字形库，简称汉字库。

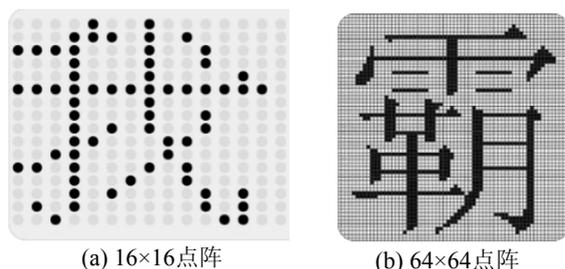


图1-7 汉字“我”和汉字“霸”的点阵表示

汉字点阵有多种规格，包括简易型 16×16 点阵、普及型 24×24 点阵、提高型 32×32 点阵、精密型 48×48 点阵和 64×64 点阵。点阵规模越大，字形也越清晰、美观，在汉字库中所占用的空间也越大。

矢量码存储的是汉字字形的轮廓特征，用数学曲线描述，字体中包含符号边界上的关键点、连线的导数信息等。输出汉字时，通过计算机的计算，描述汉字的大小和形状。矢量化字形描述与最终文字显示的大小、分辨率无关，因此可产生高质量的汉字输出。

点阵字形码使用方便、易于理解，但不同大小的字形需要不同的点阵库，占用的存储空间较大，且字形放大时容易产生锯齿状失真，优点是可以直接送到输出设备进行输出。矢量码占用的存储空间较小，字号变化时不会改变字形，效果较好，但需要进行适当处理后才能送到输出设备输出。

计算机处理汉字的一般过程(见图1-8)：输入汉字时，操作者在键盘上键入输入码，通过输入码找到汉字的区位码，再计算汉字的机内码，保存内码；显示或打印汉字时，首先从指定地址取出汉字的内码，根据内码从汉字库中取出汉字的字形码，再通过一定的转换，将字形码输出到屏幕或打印机上。

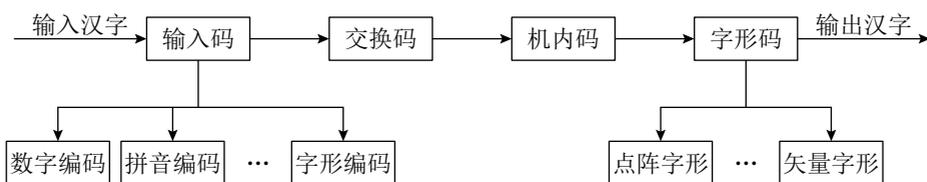


图1-8 计算机处理汉字的过程

3) Unicode码

很多传统的编码方式允许计算机支持双语环境(通常使用拉丁字母及本地语言)，却无法支持多语言环境，因此产生了Unicode码(统一码、国际码、万国码、单一码)。它为每种语言中的每个字符设定了统一并且唯一的二进制编码，以满足跨语言、跨平台进行文本转换、处理的要求，是一种可以容纳全世界所有语言文字的编码方案。

Unicode编码系统可分为编码方式和实现方式两个层次。

Unicode的编码方式规定每个字符的数字编号是多少，并不规定这个编号如何存储。Unicode的实现方式称为UTF(Unicode transformation format, Unicode转换格式)，一个字

符的Unicode编码是确定的，但是在实际传输过程中，出于跨平台及节省空间的目的，Unicode编码的实现方式有所不同。常见的UTF格式有UTF-8、UTF-16及UTF-32。自2009年以来，UTF-8一直是互联网最主要的编码形式。

UTF-8使用变长字节表示，理论上最多为6个字节，实际上其通常使用1~4字节为每个字符编码。具体编码规则如下。

- 一个ASCII字符只需要1字节编码。
- 带有变音符号的拉丁文、希腊文、西里尔字母、亚美尼亚语、希伯来文、阿拉伯文、叙利亚文等字母使用2字节编码。
- 中日韩文字、东南亚国家文字、中东国家文字等使用3字节编码。
- 其他极少使用的字符使用4~6字节编码。

UTF-8可以用来表示Unicode标准中的任何字符，而且其编码中的第一个字节仍与ASCII码相容，使得原来处理ASCII字符的软件无须或只进行少部分修改后，便可继续使用。

1.2.2 计算机系统

1. 冯·诺依曼型体系结构

当要利用计算机完成某项工作时，如完成复杂的数学计算或进行信息的管理，都必须先制定该项工作的解决方案，再将其分解成计算机能够识别并能执行的基本操作指令，最终完成程序所要实现的目标。由此可见，计算机的工作方式取决于它的两个基本能力：一是能存储程序，二是能自动执行程序。

1944年，美籍匈牙利数学家冯·诺依曼提出计算机基本结构和工作方式的设想，为计算机的诞生和发展提供了理论基础。时至今日，尽管计算机软硬件技术飞速发展，但计算机本身的体系结构并没有明显的突破，当今的计算机仍属于冯·诺依曼型体系结构，是一种将程序指令存储器和数据存储器合并在一起的计算机设计概念结构。

冯·诺依曼型计算机体系的主要思想是将程序和数据存放在计算机内部的存储器中，计算机在程序的控制下一步一步地进行处理，直到得出结果。

尽管计算机的结构有了重大变化，性能有了惊人的提高，但就结构原理来说，至今占统治地位的仍是存储程序式的冯·诺依曼型计算机体系结构，如图1-9所示。

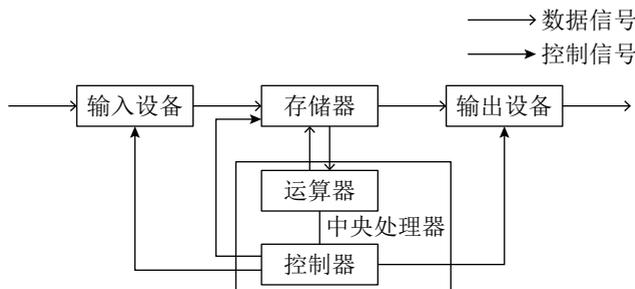


图1-9 冯·诺依曼型计算机体系结构

冯·诺依曼型计算机体系的特点如下。

1) 组成部分

冯·诺依曼型计算机由运算器、控制器、存储器、输入设备和输出设备五大部分组成，这五大部分分别完成计算机的五大功能。

(1) 运算器：完成运算功能，能完成各种算术运算、逻辑运算及数据传输等操作。

运算器又称算术/逻辑单元(arithmetic and logic unit, ALU)，是计算机处理数据、形成信息的加工厂，它的主要功能是对二进制数进行算术运算或逻辑运算。运算器主要由一个加法器、若干个寄存器和一些控制线路组成。在控制器控制下，它对取自存储器或其内部寄存器的数据进行算术或逻辑运算，其结果暂存在内部寄存器或送到存储器。

运算器的性能指标是衡量整个计算机性能的重要因素之一，与运算器相关的性能指标包括计算机的字长和运算速度。其中，字长是指计算机运算部件一次能同时处理的二进制数据的位数，字长越长，则计算机的运算速度和精度就越高。运算速度通常是指每秒所能执行加法指令的数目，常用百万次每秒(million instructions per second, MIPS)来表示，这个指标能直观地反映机器的速度。

运算器中的数据取自内存，运算结果又被送回内存。运算器在控制器的控制下对内存进行读写操作。

(2) 控制器：完成控制功能，能根据程序的规定或操作结果控制程序的执行顺序及协调计算机各部件的工作。

在计算机中，控制计算机进行某一操作的命令称为指令。控制器是计算机的神经中枢，由程序计数器、指令寄存器、指令译码器、工作脉冲形成控制电路、时钟控制信号形成部件、地址形成部件和中断控制逻辑组成。

控制器的工作过程如下：

- 从内存中取出指令，并确定下一条指令在内存中的地址。
- 对所取指令进行译码和分析，根据指令的要求向有关部件发出控制命令。
- 有关部件执行指令规定的操作。
- 将执行结果返回内存，并读取下一条指令。

从宏观上看，控制器的作用是协调计算机各部件的工作。从微观上看，控制器的作用是按一定顺序产生机器指令以获得执行过程中所需要全部控制信号，这些控制信号作用于计算机的各个部件以使其完成某种功能，从而达到执行指令的目的。所以，控制器的真正作用是对机器指令执行过程的控制。

运算器和控制器一起组成了中央处理单元，即CPU(central processing unit)，它是计算机的核心部件。中央处理单元的性能决定着整个计算机系统的性能。

(3) 存储器：完成存储功能，能记忆和保存输入的程序、数据及各种结果。

存储器是计算机中用来存储程序和数据部件，主要用于在控制器的控制下按照指定的地址存入和取出信息。它由若干存储单元组成，每个存储单元有一个编号，称为地址。

存储器分为内存储器和外存储器。

● 内存存储器简称内存(主存),是计算机的信息交流中心。用户通过输入设备输入的程序和数据最初送入内存,控制器执行的指令和运算器处理的数据取自内存,运算的中间结果和最终结果保存在内存中,输出设备输出的信息同样来自内存,内存中的信息如果需要长期保存,则应保存在外存储器中。总之,内存要与计算机各个部件打交道,进行数据传送。因此,内存的存取速度直接影响计算机的运算速度,内存容量是衡量计算机数据信息处理能力的重要标志。内存的特点是密度大、重量轻、体积小、存取速度快。

内存又分为只读存储器(ROM)和随机存储器(RAM)。ROM只能从中读取信息,而不能写入信息。当停电或死机时,其中的信息仍能保留。RAM可以从中读出和写入信息,但在断电以后将丢失存储的内容。计算机运行时,系统程序、应用程序及用户数据都临时存放在RAM中。

● 外存储器简称外存,用来存放计算机系统的系统软件、用户程序及用户的数据。通常,外存只和内存交换数据而不和计算机的其他部件直接交换数据。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CPU的输入/输出指令将其调入内存。常见的计算机存储设备有硬盘、光盘、U盘等。外存的特点是容量大、速度较慢、价格较便宜。

个人计算机中常使用三种存储技术:磁存储(如hard disk drive, HDD)、光存储(如CD、DVD)和固态存储(如各类内存卡、U盘以及solid state drive, SSD)。每种存储技术都有其优缺点,通常从通用性、耐用性、速度和容量等方面来考虑使用何种存储设备。计算机的运算器、控制器和内存存储器合称计算机的主机。

(4) 输入设备:完成输入功能,将程序和数据送到计算机的存储器中。

输入设备是人与计算机进行会话的一个接口。常用的输入设备有键盘、鼠标、扫描仪、摄像头、麦克风等。

(5) 输出设备:完成输出功能,能根据人们事先给出的格式要求,将程序、数据及结果输出给操作人员。

常用的输出设备有显示器、打印机、音箱、绘图仪等。

2) 数据和程序的存储

数据和程序以二进制代码的形式不加区别地存放在存储器中,存放位置由地址指定,地址码也为二进制形式。

3) 控制器的工作模式

控制器是根据存放在存储器中的指令序列即程序来工作的,并由一个程序计数器(PC,即指令地址计数器)控制指令的执行。控制器有判断能力,能根据计算结果选择不同的动作流程。

2. 计算机的工作原理和基本结构

计算机硬件系统的各大部件并不是孤立存在的,在处理信息的过程中,各部件需要互相传输数据,部件之间基本上都有单独的连接线路。计算机的工作原理和基本结构如图1-10所示。

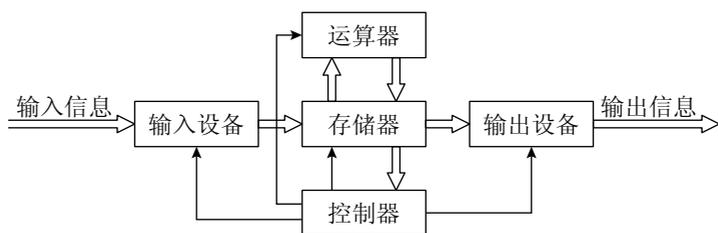


图1-10 计算机的工作原理和基本结构

计算机操作系统启动后，输入设备处于等待用户输入数据的状态，用户输入时，输入设备向控制器发出输入请求，控制器向输入设备发出输入命令，用户将编写的源程序、命令及各种数据通过输入设备传送到内部存储器中，依次执行输入的命令或程序指令，控制器发出存取命令，数据存入内部存储器或从内部存储器中取出数据；控制器根据程序指令的运算请求发出取数据命令，从内部存储器中取数据送入运算器的缓冲器参与运算，运算的结果保存到内部存储器中。当程序需要输出时，控制器通知输出设备，输出设备准备好后向控制器发送输出请求，控制器发送输出命令，数据从内部存储器传送到输出设备，输出运行结果。

3. 计算机系统的组成

计算机系统由硬件系统和软件系统两大部分组成，如图1-11所示。硬件是指由电子线路、元器件和机械部件等构成的具体装置，是看得见、摸得着的实体，是机器系统。软件系统是计算机中运行的程序及其使用的数据和相应文档的集合。没有软件系统的计算机几乎是没有用的。计算机的功能不仅取决于硬件系统，在更大程度上是由所安装的软件系统所决定的。

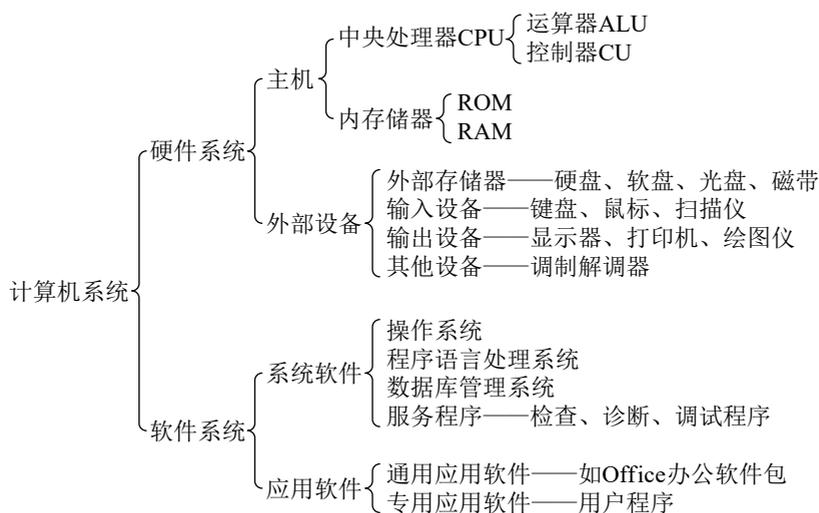


图1-11 计算机系统的组成

4. 微型计算机的硬件系统

微型计算机是指以微处理器为中心，同时配置相应的主存储器、输入/输出接口电

路、系统总线和总线接口，以及相应的外围设备的计算机系统。

微型计算机的硬件系统主要由系统主板、CPU、存储器，以及各种输入、输出设备组成。

1) 系统主板

系统主板又称主板，用于连接计算机的多个部件。它安装在主机箱内，是微型计算机最基本、最重要的部件之一。主板主要包括CPU插槽、内存插槽、芯片组、总线扩展槽、BIOS、各种接口。目前主板一般都集成了显卡、声卡、网卡、无线网卡等。

(1) CPU插槽。CPU插槽用于固定连接CPU的芯片。

(2) 内存插槽。用户购买与主板插槽匹配的内存就可以实现内存扩充，即插即用。

(3) 芯片组。芯片组是主板的灵魂，由一组超大规模集成电路芯片构成。芯片组控制和协调整个微机系统的正常运转和各个部件的选型，它被固定在主板上，不能像CPU、内存等进行简单的升级换代。芯片组的作用是在BIOS和操作系统的控制下，按照统一的技术标准和规范为计算机中的CPU、内存、显卡等部件建立可靠的安装、运行环境，为各种接口的外部设备提供可靠的连接。芯片组的外观就是集成块。目前，芯片组的生产厂家主要有Intel、VIA、AMD、NVIDIA、ATI等，其中Intel和VIA的芯片组最为常见。

(4) 总线扩展槽。总线扩展槽主要用于扩展微型计算机的功能，也称为I/O插槽，可以插入许多标准选件，如显卡、声卡、网卡等，以扩展微型计算机的各种功能。任何插卡插入扩展槽后，都可以通过系统总线与CPU连接，在操作系统的支持下实现即插即用。这种开放式结构方便用户对微机相应子系统进行局部升级，使厂家和用户配置机型方面有更大的灵活性。

(5) BIOS。BIOS即基本输入/输出系统(basic input/output system)，是主板的核心，它保存着计算机系统中的基本输入/输出程序、系统信息设置、自检程序等，并反馈设备类型、系统环境等信息。现在的BIOS芯片中还加入了电源管理、CPU参数调整、系统监控、PNP(即插即用)、病毒防护等功能。

(6) 各种接口。接口是指计算机系统中，在两个硬件设备之间起连接作用的逻辑电路。接口的功能是在各个组成部件之间进行数据交换。主机与外部设备之间的接口称为输入/输出接口，简称I/O接口。

- 集成设备电子部件(integrated device electronics, IDE)接口，主要连接IDE硬盘和IDE光驱。主板上有两组IDE设备接口，分别为IDE1和IDE2。IDE1通常用于连接引导硬盘，IDE2多用于接入光驱。

- 串行接口(serial port)简称串口。微型机中采用串行通信协议的接口称为串行接口，也称为RS-232接口。

- 并行接口(parallel port)简称并口，用一组线同时传送几组数据。在微型机中，一般配置一个并行接口，标记为LPT1或PRN。并口一般用于连接老式的打印机，目前很多主板已经不提供并口了。

- USB(universal serial bus)接口即符合通用串行总线硬件标准的接口,用于外部设备。USB能使相关外设机箱外连接,允许热插拔(连接外设时不必关闭电源),实现安装自动化,且比传统串口快成百上千倍。各类设备(如鼠标、键盘、打印机、扫描仪等)均已转为使用USB接口。

2) CPU

CPU是一个大规模集成电路芯片,包括运算器、控制器、寄存器组、内部总线等。寄存器用于暂存参与运算的数据、结果和状态等,高档CPU芯片中还有高速缓冲存储器,用于解决CPU与内存之间速度不匹配的问题。

CPU的性能指标主要包括两个:机器字长和主频。

机器字长是指计算机的运算部件能同时处理的二进制数据的位数。字长决定了计算机的运算精度,字长越长,计算机的运算精度就越高。因此,高性能的计算机字长较长,而性能较差的计算机字长相对要短一些。字长也影响计算机的运算速度,字长越长,计算机在一个周期内处理的数据位数就越多,运算速度就越快。字长通常是字节的整倍数,如Intel奔腾系列CPU字长为32位,而酷睿2系列CPU字长达到64位。近几年,主流CPU为64位字长,处理器一次可以运行64位数据。64位计算机主要有两大优点:可以进行更大范围的整数运算,可以支持更大的内存。

主频即计算机CPU的时钟频率,又称时钟周期和机器周期,单位是兆赫(MHz)或吉赫(GHz),它反映了CPU的基本工作节拍。例如,2.4GHz主频是指CPU时钟能在1s内运行2.4亿个周期。周期是CPU的最小时间单位。CPU的每一项活动都以周期来度量。需要注意的是,时钟的速度并不等于处理器在1s内能执行的指令数目。在很多计算机中,一些指令能在1个周期内完成,但是也有一些指令需要多个周期才能完成。有些CPU甚至能在单个时钟周期内执行几个指令。

主频是衡量CPU性能的一个重要技术指标。主频越高,表明指令的执行速度越快,指令的执行时间也就越短,对信息的处理能力和效率就越高。

注意,只有在比较同系列芯片的CPU时,才可以直接对时钟速度加以比较。

另外,对多核处理器来说,核心的数量也会影响性能。双核(多核)结构就是在一个CPU中集成两个(多个)单独的CPU单元。这种技术的好处是可以在一个时钟周期内执行多条指令,因而理论上可以成倍提高CPU的处理能力。多核心通常会带来更快的处理速度。2.4GHz的i5处理器有两个核心,等效性能为4.8GHz($2.4\text{GHz} \times 2$)。而1.6GHz的i7处理器有4个核心,等效性能为6.4GHz($1.6\text{GHz} \times 4$)。

如果是不同架构、不同品牌的CPU,仅对比主频没有可比性,还需要了解CPU的核心架构、核心数量及缓存容量等几个重要指标。

3) 存储器

存储器是用来存储程序和数据部件。从存储器中取出信息而不破坏原有的内容,这种操作称为存储器的读操作;把信息写入存储器而将原来的内容抹掉,这种操作称为存储器的写操作。存储器分为内存(又称主存储器)和外存(又称辅助存储器)两大类。

内存与运算器和控制器直接相连，存放当前正在运行的程序和有关数据，存取速度较快；外存存放计算机暂时不用的程序和数据，需要时才调入内存，它的存取速度相对较慢。通常，运算器、控制器、主存储器合称为计算机的主机。

4) 输入、输出设备

输入设备(如键盘、鼠标等)能把程序、数据、图形、声音、控制现场的模拟量等信息，通过输入接口转换为计算机可接收的形式。输出设备(如显示器、打印机等)能把计算机的运行结果或过程，通过输出接口转换为人们所要求的直观形式或控制现场能接受的形式。而不少设备同时集成了输入、输出两种功能。例如，光盘刻录机可作为输入设备，将光盘上的数据读入计算机内存，也可作为输出设备将数据刻录到CD-R或CD-RW光盘上。

5. 计算机的软件系统

计算机的软件系统是指为支持计算机运行或解决某些特定问题而需要的程序、数据以及相关文档的集合。

计算机的硬件系统也称为裸机，裸机只能识别由0和1组成的机器代码。没有软件系统的计算机是无法工作的，它只是一台机器而已。实际上，用户所面对的是经过若干层软件“包装”的计算机，计算机的功能不仅仅取决于硬件系统，在更大程度上是由所安装的软件系统决定的。硬件系统和软件系统互相依赖，不可分割。计算机硬件、软件与用户之间的关系是一种层次结构，其中硬件处于内层，用户在最外层，而软件则在硬件与用户之间，用户通过软件使用计算机的硬件。计算机系统的层次结构如图1-11所示。

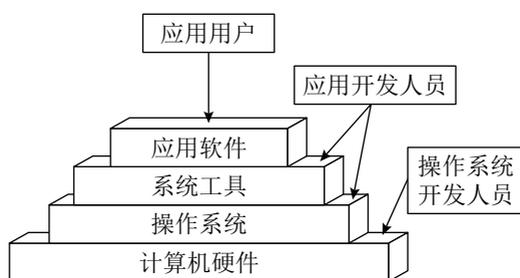


图1-11 计算机系统的层次结构

1) 软件的定义

软件是计算机的灵魂，没有软件的计算机毫无用处。软件是用户与硬件之间的接口，用户通过软件使用计算机硬件资源。计算机科学对软件的定义：软件就是在计算机系统支持下，能够完成特定功能和性能的程序、数据及相关的文档。于是，软件可以形式化地表示为

$$\text{软件} = \text{知识} + \text{程序} + \text{数据} + \text{文档}$$

程序是用计算机程序设计语言描述的。无论是低级语言(如汇编语言)，还是高级语言(如C++、Java、Python等)，程序都可以在相应的语言编译器的支持下转换为操纵计算机硬件执行的代码。

2) 软件的特点

- (1) 具有抽象性，是一种逻辑实体，只能通过运行状况来了解其功能、特性和质量。
- (2) 软件没有明显的制作过程。
- (3) 软件不存在磨损、老化问题，但存在缺陷需要维护和技术更新等问题。
- (4) 软件的开发和运行必须依赖于特定的计算机系统环境，对硬件有依赖性，为了减少依赖，开发中提出了软件的可移植性。
- (5) 复杂性高、成本高，软件的开发渗透了大量的脑力劳动，以及人的逻辑思维、智能活动和技术水平。
- (6) 软件的开发涉及诸多社会因素，如知识产权等。

3) 软件的分类

计算机软件分为系统软件(system software)和应用软件(application software)两大类。

(1) 系统软件。系统软件是管理、监控和维护计算机资源的软件，是用来扩大计算机的功能、提高计算机的工作效率、方便用户使用计算机的各种程序的集合。人们借助软件使用计算机。系统软件是计算机正常运转不可缺少的，一般在购买计算机时由厂家提供。任何用户都要用到系统软件，其他程序都要在系统软件的支持下运行。系统软件分为操作系统、语言处理系统、数据库管理系统及系统工具软件4类。

① 操作系统。操作系统(operating system, OS)是用户和计算机硬件之间的操作平台，用户只有通过操作系统才能在不了解计算机系统内部结构的情况下正确使用计算机。所有的应用软件和其他的系统软件都是在操作系统下运行的。目前使用的操作系统有很多不同的版本，其功能各具特色，适用于不同的场合。目前在微机上运行的操作系统主要有MS-DOS、Windows、macOS、UNIX、Linux等，在手持计算机上运行的操作系统主要有Android、iOS等。

② 语言处理系统。语言处理系统将高级语言编写的源程序翻译成由机器语言(一种以二进制代码“0”和“1”形式来表示，且能够被计算机直接识别和执行的语言)组成的目标程序。高级语言贴近自然语言，用户不必了解计算机的内部结构，只需要把解决问题的执行步骤输入计算机即可。高级语言不能被计算机执行，必须通过语言处理系统转换为计算机能够识别的机器语言，其转换过程有如下两种方式。

- 编译方式：编译程序把高级语言的源程序翻译成用机器指令生成的目标程序，再由计算机执行该目标程序并得到计算结果。
- 解释方式：解释程序对源程序逐句地进行翻译，每翻译一句就由机器执行一句，即边解释边执行。

不同的高级语言有不同的语言处理系统。C++等高级语言源程序采用编译方式，而Python等则采用解释方式。

③ 数据库管理系统。数据库管理系统(database management system, DBMS)是进行数据存储、共享和处理的有效工具。当今计算机已广泛应用于各种管理工作中，而完成这种

管理工作的信息管理系统几乎都是以数据库为核心的。简单地说，数据库管理系统是管理系统中大量、持久、可靠、共享的数据的工具。数据库管理系统在操作系统支持下工作。常见的数据库管理系统有Access、Oracle、DB2、SQL Server、MySQL等。

④ 系统工具软件。系统工具软件是指用来管理、维护、使用计算机的服务性程序，如诊断和修复工具、调试程序、编辑程序、文件压缩程序和磁盘整理工具等，主要是为了维护计算机系统正常运行，方便用户进行软件开发和运行，如Windows中的磁盘整理工具程序等。还有一些著名的工具软件，如360软件管家，它集成了维护计算机的各种工具程序。实际上，Windows和其他操作系统都有附加的实用系统工具程序。

(2) 应用软件。应用软件是为解决某一特定应用领域内的任务而开发的软件，是在系统软件的支持下工作的。常用的应用软件如下。

① 通用应用软件。通用应用软件由计算机专业人员与相关专业的技术人员共同开发完成，是为解决通用性问题而研制开发的程序，通常由软件开发商发布与发行，使用范围较广，如文字处理软件Word、WPS，电子表格软件Excel，绘图软件AutoCAD、网络浏览软件Internet Explorer、Chrome、Firefox等。

② 专用应用软件。专用应用软件也称用户程序，是指用户自行开发或者委托软件企业开发的针对特定问题而编制的程序。专用应用软件专门用于某一个专业领域，如股票分析软件、银行管理软件、气象预报分析系统、企业财务管理系统、仓库管理系统、人事档案管理系统、设备管理系统、计划管理系统等，以及广泛使用的各种管理信息系统(management information system, MIS)。

1.2.3 操作系统和文件

1. 操作系统

操作系统是计算机系统中的一个系统软件，它们管理和控制计算机系统中的软件和硬件资源，合理地组织计算机工作流程，以便有效地利用这些资源为用户提供一个功能强大、使用方便和可扩展的工作环境，从而在计算机与其用户之间起到接口的作用。

操作系统的目标主要有两点：一是方便用户使用计算机，一个好的操作系统应提供给用户一个清晰、简洁、易于使用的用户界面；二是提高系统资源的利用率，尽可能使计算机系统中的各种资源得到最充分的利用。

1) 操作系统的功能

操作系统的主要功能可以分为处理机管理、存储管理、设备管理、文件管理和作业管理等。

(1) 处理机管理。处理机管理主要是对CPU进行管理，又称进程管理。CPU是计算机系统最重要的硬件资源，计算机的一切处理和运算都是在CPU中完成的。处理机的占用率和它的利用率直接关系计算机和用户任务的处理效率。

CPU的每个周期都是可用于完成任务的资源。许多被称为“进程”的计算机活动会竞争CPU的资源。进程是指正在执行的程序，即进程=程序+执行。进程是程序的一次执行过程，是系统进行调度和资源分配的一个独立单位。一个程序被加载到内存，系统就创建了一个进程，程序执行结束后，该进程也就消亡了。当有一个或多个用户提交作业请求服务时，操作系统对进程的管理是协调各作业之间的运行，充分发挥CPU的作用，为所有用户服务，提高计算机的使用效益，使CPU的资源得到充分利用。

在Windows等操作系统中，用户可以查看当前正在执行的进程。有时“进程”又称“任务”。例如，在Windows任务管理器(按Alt+Ctrl+Delete组合键打开)中，可以快速查看进程信息，或者强行中止某个进程。当然，结束一个应用程序的最好方式是在应用程序的界面中正常退出，而不是在进程管理器中删除一个进程，除非应用程序出现异常而不能正常退出时才这样做。如果怀疑程序没有正确关闭或有恶意软件在暗中捣鬼，用户可以查看CPU正在执行哪些进程。

(2) 存储管理。存储管理的主要任务是对内存资源进行合理分配。当多个程序共享有限的内存资源时，如何为它们分配内存空间，使它们既彼此隔离、互不干扰，又能在一定条件下及时调配，尤其是内存不够用时，如何把当前未运行的程序与程序所需数据及时调出内存，运行时再从外存调入内存等，都是存储管理的任务。

(3) 设备管理。设备管理是指计算机系统中，CPU和内存以外的所有输入/输出设备的管理，不仅包括进行实际输入/输出操作的设备，还包括各种支持设备。设备管理的首要任务是为这些设备提供驱动程序或控制程序，使用户不必了解设备及接口技术细节，就可方便地对这些设备进行操作。另外，设备管理会协调相对低速的外部设备尽可能与CPU并行工作，以提高设备的使用效率和整个系统的运行速度。

(4) 文件管理。文件是具有某种性质的信息集合。文件的范围很广，包括文本文件、程序文件、应用文件等。文件通常存放在外存(如磁盘)上，通过文件名即可对文件的内容进行读写操作。文件是计算机系统的软件资源，有效地组织、存储、保护文件，使用户方便、安全地访问它们，是文件管理的任务。

(5) 作业管理。所谓作业，就是在一次提交给计算机处理的程序和数据的集合或一次事务处理中，要求计算机系统所做的工作的集合。可以说，计算机的一切工作都是为了完成作业。协助用户向计算机系统提交作业，保障系统以较高的效率运行，这就是作业管理的任务。

2) 操作系统的演变与发展

操作系统伴随着计算机技术及其应用的发展而逐渐发展和不断完善，其功能由弱到强，在计算机系统中的地位不断提高，至今，已成为计算机系统的核心。操作系统的发展历史如下。

(1) 无操作系统时代。1946年，世界上第一台电子计算机ENIAC诞生，计算机硬件主要采用电子管器件，输入、输出等各类操作命令均由手工实现。

(2) 第一代操作系统。20世纪50年代初期，产生了第一个简单的批处理操作系统，即

操作系统的雏形——批处理系统(监督程序),用来控制作业的运行。用户将作业提交到机房,操作员将一批作业输入外存(如磁带),形成一个作业队列。当需要调入作业时,监督程序从这一批作业中选择一项调入内存运行。当这一项作业完成时,监督程序再调入另一项作业,直到这一批作业全部完成。

(3) 第二代操作系统。20世纪60年代中期,产生了多道操作系统、分时操作系统。

多道操作系统:在计算机内存中同时存放几个相互独立的程序,在管理程序控制之下,使它们在系统内相互穿插地运行。

分时操作系统:在一台主机上连接多个带有显示器和键盘的终端,同时允许多个用户通过自己的键盘,以交互的方式使用计算机,共享主机中的资源。

(4) 第三代操作系统。20世纪70年代,通用计算机操作系统开始出现,如UNIX、MS-DOS等操作系统相继问世。

UNIX系统自诞生至今已50多年,仍然是PC、服务器、中小型机、工作站、巨型机及集群等类型的计算机的通用操作系统,而且以其为基础形成的开放系统标准(如POSIX)也是迄今为止唯一的操作系统标准。

MS-DOS是Microsoft disk operating system的简称,是由美国微软公司提供的单用户磁盘操作系统,从4.0版开始具有多任务处理功能。

(5) 第四代操作系统。20世纪八九十年代以后,出现了Windows系列操作系统、网络操作系统、分布式操作系统等。

Windows系列操作系统自1985年推出Windows 1.0以来不断推陈出新,因其易学、易用、友好的图形用户界面,以及多任务及内存扩展等功能得以很快流行并迅速占领市场,至今仍是操作系统的主流产品。

网络操作系统能够管理网络通信和网络上的共享资源,协调各个主机上任务的运行,并向用户提供统一、高效、方便、易用的网络接口。UNIX、Linux及用于服务器的Windows版本都是网络操作系统。

分布式操作系统是指由多个分散的处理单元经网络的连接而形成的系统。在分布式操作系统中,系统的处理和控制功能都分散在系统的各个处理单元上,系统中的所有任务都可以动态地分配到各个处理单元中。分布式操作系统是网络操作系统的更高形式。

(6) 第五代操作系统。第五代操作系统与硬件结合更加紧密,嵌入式操作系统(embedded operating system, EOS)及移动操作系统均是这一代操作系统的代表。

嵌入式系统是一种完全嵌入受控器件内部,为特定应用而设计的专用计算机系统。它结构精简,在硬件和软件上都只保留需要的部分,而将不需要的部分裁去。一般都具有便携、低功耗、性能单一等特性,在智能家居、交通管理、环境监测、电子商务等领域的智能终端上均有广泛的应用。嵌入式操作系统是指用于嵌入式系统的操作系统,是一种用途广泛的系统软件,例如嵌入式Linux、WinCE,负责嵌入式系统的全部软硬件资源的分配、调度工作,控制与协调并发活动。

另外,移动操作系统也随着智能手机和平板计算机的不断发展而发展起来,主流的移

动操作系统有苹果的iOS和 Google的 Android等。

2. 文件

文件是具有文件名的相关信息的集合，所有的程序和数据都是以文件的形式存放在计算机的外存上。想有效地使用计算机文件，就需要对文件的基础知识有很好的理解。

1) 文件名和扩展名

任何一个文件都有文件名，文件名是存取文件的依据，即“按名存取”。保存文件时，必须提供符合特定规则的有效文件名，这些特定规则称为文件命名规范。

用户给文件命名时，必须遵循以下规则：

(1) 在文件和文件夹的名称中，用户最多可使用255个字符。

(2) 可使用带有多个间隔符“.”的文件名，如jsj.jsj.docx。

(3) 文件名可以有空格但不能有“\”“/”“:”“*”“?”“|”“<”“>”“|”等。

(4) Windows保留文件名的大小写格式，但不能区分大小写，例如JSJ.TXT 和jsj.txt被认为是同一文件名。

(5) 搜索和显示文件时，用户可以使用通配符“?”和“*”。其中，问号“?”代表一个任意字符，星号“*”代表任意个字符。

(6) 文件名中最后一个“.”后的字符串被称为扩展名，用以标识文件类型，例如jsj.jsj.docx的扩展名为docx，表示该文件是一个Word文档。

在Windows 10系统的平铺显示方式下，文件主要由文件名、文件扩展名、分隔点、文件图标及文件描述信息等部分组成，如图1-12所示。

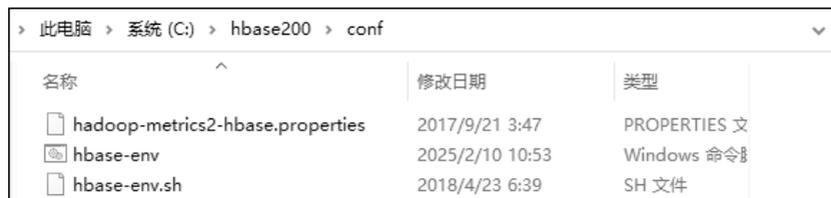


图1-12 文件的组成

常用的Windows文件扩展名及其表示的文件类型如表1-7所示。

表1-7 常用的 Windows 文件扩展名及其表示的文件类型

扩展名	文件类型	扩展名	文件类型
AVI	视频文件	FON	字体文件
BAK	备份文件	HLP	帮助文件
BAT	批处理文件	INF	信息文件
BMP	位图文件	MID	乐器数字接口文件
COM	执行文件	MMF	Mail文件
DAT	数据文件	RTF	文本格式文件

(续表)

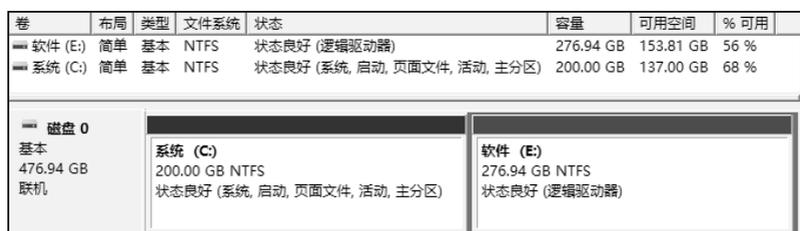
扩展名	文件类型	扩展名	文件类型
DCX	传真文件	SCR	屏幕文件
DLL	动态链接库	TTF	TureType字体文件
DOC	Word 文件	TXT	文本文件
DRV	驱动程序文件	WAV	声音文件

可以通过设置来显示或隐藏文件的扩展名。设置显示和隐藏文件扩展名的方法：首先打开“此电脑”或“文件资源管理器”窗口，选择“查看”→“选项”命令，在弹出的“文件夹选项”对话框中选择“查看”选项卡，在“高级设置”中，取消勾选“隐藏已知文件类型的扩展名”复选框即可显示文件扩展名。

2) 文件夹和路径

要指定文件的位置，首先必须指定文件存储在哪个设备中。在Windows操作系统中，个人计算机的每一个存储设备都是以驱动器名(也叫盘符)进行识别的。盘符通常由图标、名称和信息组成，用大写字母加一个冒号来表示，如“C:”，简称C盘。

在Windows中硬盘驱动器被指定为“C:”，用户可以根据自己的需求创建多个硬盘分区，在不同的磁盘上存放相应的内容，一般来说，C盘是第一个磁盘分区，用来存放系统文件，程序和数据可以存放在其他分区。各个磁盘在计算机中的显示状态如图1-13所示。



卷	布局	类型	文件系统	状态	容量	可用空间	% 可用
软件 (E:)	简单	基本	NTFS	状态良好 (逻辑驱动器)	276.94 GB	153.81 GB	56 %
系统 (C:)	简单	基本	NTFS	状态良好 (系统, 启动, 页面文件, 活动, 主分区)	200.00 GB	137.00 GB	68 %

磁盘 0	系统 (C:)	软件 (E:)
基本 476.94 GB 联机	200.00 GB NTFS 状态良好 (系统, 启动, 页面文件, 活动, 主分区)	276.94 GB NTFS 状态良好 (逻辑驱动器)

图1-13 各个磁盘在计算机中的显示状态

操作系统为每个存储设备维护一个称为目录的文件列表。主目录也称根目录。根目录通过驱动器名加反斜杠来表示，如“C:\”表示硬盘的一个根目录。根目录还可以进一步细分为更小的列表，一个列表称为一个子目录。

在Windows中，子目录即文件夹，简单地说，文件夹就是文件的集合，类似于文件柜中存放的相关文件的文件夹。如果计算机中的文件过多，则会显得杂乱无章，不方便查找文件，此时用户可将相似类型的文件整理起来，统一地放置在一个文件夹中，这样不仅可以方便用户查找文件，还能有效地管理计算机中的资源。

文件夹中可以包含文件和子文件夹，子文件夹中又可以包含文件和子文件夹，以此类推，即可形成文件和文件夹的树形关系。文件夹中可以包含多个文件和文件夹，也可以不包含任何文件和文件夹。不包含任何文件和文件夹的文件夹称为空文件夹。文件夹名称及图标如图1-14所示。

名称	修改日期	类型	大小
bin	2025/2/10 11:39	文件夹	
conf	2025/2/10 11:39	文件夹	
docs	2025/2/10 11:40	文件夹	
hbase-webapps	2025/2/10 11:40	文件夹	
lib	2025/2/10 11:40	文件夹	
CHANGES.md	2019/3/20 2:58	MD 文件	95 KB
LEGAL	2019/3/20 2:58	文件	1 KB
LICENSE	2019/3/20 7:13	文本文档	127 KB

图1-14 文件夹名称及图标

路径指文件或文件夹在计算机中存储的位置，打开某个文件夹，即可在地址栏中看到进入的文件夹的层次结构，地址栏如图1-15所示。由文件夹的层次结构可以得到文件夹的路径。

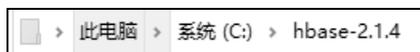


图1-15 地址栏

路径的结构包括盘符、文件夹名称和文件名称，它们之间用“\”隔开。例如C盘下“hbase-2.1.4”文件夹里的“license.txt”文件，路径显示为C:\hbase-2.1.4\license.txt。

3) 文件格式

文件格式是指存储在文件中的数据组织和排列方式。显然，音乐文件的存储方式与文本文件和图形文件的存储方式是不同的，甚至对同一类数据，也有很多不同的文件格式，例如，图形数据可存储为BMP、GIF、JPEG或PNG等文件格式。对程序、文字、图片等每一类信息，都可以以一种或多种文件格式保存到计算机中。每一种文件格式通常会有一种或多种扩展名来识别，但也可以没有扩展名。扩展名可以帮助应用程序识别文件格式。

每一种应用软件都可以处理特定的文件格式。打开“打开”对话框时，多数应用程序会自动筛选文件，只显示那些以它们能处理的文件格式存储的文件。在Windows中，可使用文件关联列表把文件格式和相应的应用软件连接起来，以便用户双击某个文件名时，计算机可以自动打开能处理正确文件格式的应用软件。

此外，可对文件格式进行转换。例如，若某人创建了一个Word文档，想将其发布到Web上，则可以将此文档的格式转换为HTML格式后发布。最简单的文件格式转换方法就是找到一种能处理这两种文件格式的应用软件，然后使用该软件打开需要转换格式的文件，使用“导出”或“另存为”对话框选择一种新的文件格式，给这个文件重新命名后保存。

本章小结

本章主要介绍了计算机的基础知识和信息技术的发展历程，首先阐述了计算机的定

义、核心特征以及其在现代社会中的重要地位，随后详细回顾了计算机从起源到现代的发展历程，包括电子管、晶体管、集成电路到超大规模集成电路的四个主要阶段。在理解计算机硬件系统的基础上，本章探讨了计算机的软件系统，包括系统软件和应用软件的分类与作用。此外，本章介绍了计算机信息处理的基本原理，包括二进制编码、数值编码和字符编码等核心概念，以及计算机在科学计算、数据处理、过程控制和人工智能等多个领域的广泛应用。读者通过学习本章，可以对计算机的基础知识和信息技术的发展历程有全面而深入的了解。

习题

1. 简答题

- (1) 简述计算机的五大组成部分及其功能。
- (2) 解释冯·诺依曼型计算机体系结构的主要特点。
- (3) 什么是二进制？为什么计算机使用二进制进行数据处理？
- (4) 简述ASCII码和Unicode码的区别。

2. 计算题

- (1) 将十进制数123转换为二进制数和十六进制数。
- (2) 将二进制数1101011转换为十进制数和八进制数。
- (3) 计算二进制数的和： $1011 + 1101$ 。
- (4) 将十六进制数A3F转换为二进制数和十进制数。

3. 应用题

- (1) 假设一个文件大小为2GB，计算其对应的字节数，分别以KB、MB为单位进行计算。
- (2) 解释计算机中“存储程序”的概念，并举例说明其在现代计算机中的应用。
- (3) 简述计算机的发展历程，列举每一代计算机的主要特点。
- (4) 解释计算机中“字长”的概念，并说明其对计算机性能的影响。

4. 编程题

- (1) 编写一个Python程序，将用户输入的十进制数转换为二进制数和十六进制数。
- (2) 编写一个Python程序，计算两个二进制数的和，并输出结果。