



全国电子信息优秀教材
普通高等教育“十三五”规划教材
新工科建设之路·计算机类规划教材

大学C/C++语言程序设计基础

(第3版)

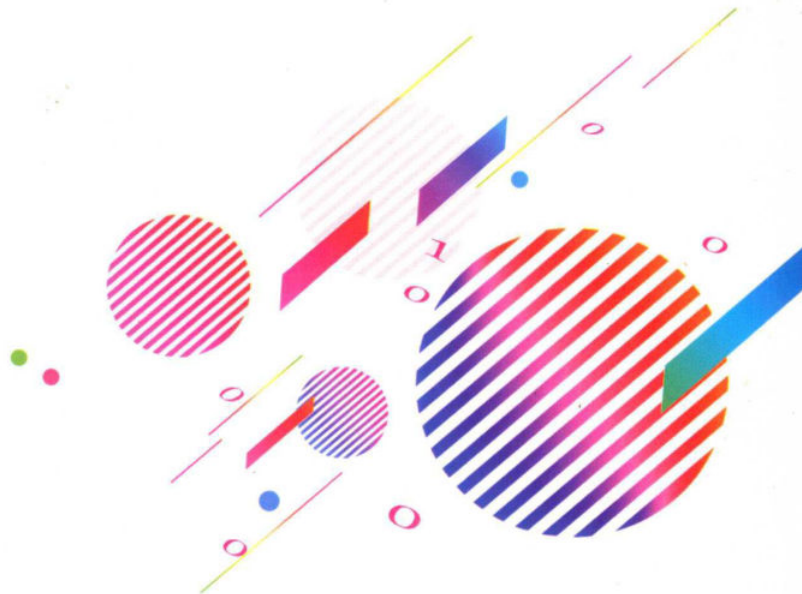
◎ 阳小华 李晓昀 马淑萍 主编
◎ 刘志明 主审



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>



大学C/C++语言程序设计基础

(第3版)

本书在第2版的基础上修订而成，以计算思维为主线重新组织内容，同时强调掌握科学计算工具和培养科学计算能力对理工类学生的重要性。在绪论中介绍了当今计算机前沿技术，如大数据、云计算与边缘计算、人工智能、信息检索等相关内容；增加了计算机系统的组成、工作原理、存储机制、数制、编码、信息数字化等学习程序设计的先导知识。全书系统介绍了C/C++语言及科学计算软件MATLAB的基本概念和语法规则。

全书共12章，主要内容包括：绪论、C语言与MATLAB基础、数据的输入/输出、选择结构程序设计、循环结构程序设计、函数与编译预处理、数组、指针、构造数据类型、文件、C++面向对象程序设计基础、C/C++与MATLAB混合编程。附录中列出了C语言常用库函数和MATLAB函数表。本书还设计了丰富的实例，以增强内容的实用性和可读性，提高学生的编程兴趣。本书兼顾全国计算机等级考试的要求。为方便教学，本书配有电子课件和相关程序源代码，任课教师可以登录华信教育资源网（www.hxedu.com.cn）免费注册下载。

本书可作为高等学校理工类非计算机专业的程序设计教材，也可作为全国计算机等级考试的辅助教材，还可供程序设计爱好者参考。

相关图书：《大学C/C++语言程序设计实验教程》（第3版）

阳小华 邹腊梅 胡义香 主编 刘志明 主审

ISBN 978-7-121-37074-8

ISBN 978-7-121-37075-5



9 787121 370755 >

定价：59.00 元



责任编辑：戴晨辰
封面设计：张 昱

全国电子信息优秀教材
普通高等教育“十三五”规划教材
新工科建设之路·计算机类规划教材

大学 C/C++ 语言程序设计基础

(第3版)

阳小华 李晓昀 马淑萍 主 编
刘志明 主 审

电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书在第2版的基础上修订而成，以计算思维为主线重新组织内容，同时强调掌握科学计算工具和培养科学计算能力对理工类学生的重要性。在绪论中介绍了当今计算机前沿技术，如大数据、云计算与边缘计算、人工智能、信息检索等相关内容；增加了计算机系统的组成、工作原理、存储机制、数制、编码、信息数字化等学习程序设计的先导知识。全书系统介绍了C/C++语言及科学计算软件MATLAB的基本概念和语法规则。

全书共12章，主要内容包括：绪论、C语言与MATLAB基础、数据的输入/输出、选择结构程序设计、循环结构程序设计、函数与编译预处理、数组、指针、构造数据类型、文件、C++面向对象程序设计基础、C/C++与MATLAB混合编程。附录中列出了C语言常用库函数和MATLAB函数表。本书还设计了丰富的实例，以增强内容的实用性和可读性，提高学生的编程兴趣。本书兼顾全国计算机等级考试的要求。为方便教学，本书配有电子课件和相关程序源代码，任课教师可以登录华信教育资源网（www.hxedu.com.cn）免费注册下载。

本书可作为高等学校理工类非计算机专业的程序设计教材，也可作为全国计算机等级考试的辅助教材，还可供程序设计爱好者参考。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

大学C/C++语言程序设计基础/阳小华，李晓昀，马淑萍主编. —3版. —北京：电子工业出版社，2019.9
ISBN 978-7-121-37075-5

I. ①大… II. ①阳… ②李… ③马… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆CIP数据核字（2019）第144521号

责任编辑：戴晨辰

印 刷：湖北画中画印刷有限公司

装 订：湖北画中画印刷有限公司

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：22.5 字数：590.4千字

版 次：2011年3月第1版

2019年9月第3版

印 次：2019年9月第1次印刷

定 价：59.00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：dcc@phei.com.cn。

前 言

为了把“计算思维能力的培养”作为计算机基础教育的核心任务，本书在第2版的基础上进行完善，内容更加与时俱进，介绍了当今计算机前沿技术，如大数据、云计算与边缘计算、人工智能、信息检索等相关内容；增加了计算机系统的组成、工作原理、存储机制、数制、编码、信息数字化等学习程序设计的先导知识。全书系统地介绍了 C/C++ 语言及科学计算软件 MATLAB 的基本概念和语法规则。

C/C++ 语言是学习程序设计的第一门语言和专业必修的编程语言，不仅因为其结构严谨、数据类型完整、语句简练灵活、运算符丰富，更因为很多高级语言都是在它的基础上发展起来的。学好 C/C++ 语言对于开发底层程序及高效的程序都很有帮助。

鉴于理工类学生在本科的学习及今后的工作中会大量使用运算，其中包括矩阵运算、曲线拟合、数据分析等，因此本书除讲解传统 C/C++ 语言程序设计外，还介绍了代表当今科学计算软件先进水平的 MATLAB 软件，并增加了工程计算实例，让读者通过 C/C++ 语言编程对这类大型软件中的某些功能进行实现，旨在提醒学生掌握科学计算工具和培养科学计算能力的重要性。

开设 C/C++ 语言程序设计课程的目的不是单纯教会学生利用一种计算机语言编程，而是使学生了解信息技术的发展现状，学会信息资源的获取及筛选方法；使学生学会利用计算机程序设计来进行问题求解，学会使用科学的计算工具，具备科学的计算能力，进行数据分析和数据处理；培养学生的计算思维能力、严谨的工作作风和团队合作精神；提高学生对专业领域技术问题的理解，提高归纳总结和提出问题的能力，为进一步的专业学习、自主学习和终身学习打下基础。

全书共 12 章，主要内容包括：绪论、C 语言与 MATLAB 基础、数据的输入/输出、选择结构程序设计、循环结构程序设计、函数与编译预处理、数组、指针、构造数据类型、文件、C++ 面向对象程序设计基础、C/C++ 与 MATLAB 混合编程。附录中列出了 C 语言常用库函数和 MATLAB 函数表。

本书在编写时兼顾了全国计算机等级考试的要求。书中例题丰富，注重实用性。

本书与《大学 C/C++ 语言程序设计实验教程》（第 3 版）（阳小华、邹腊梅、胡义香主编，电子工业出版社出版，ISBN 978-7-121-37074-8）配套使用。为方便教师教学和学生学习，本书提供电子课件和程序源代码，读者可以登录华信教育资源网（www.hxedu.com.cn）免费注册下载。

本书由阳小华、李晓昀、马淑萍主编；全书由刘志明主审；邹腊梅、胡义香、熊东平、汪凤麟参加了本书的编写。由于编写时间仓促，加之作者水平有限，书中难免有错误和不妥之处，恳请各位读者和专家批评指正，以便再版时及时修正。

目 录

第 1 章 绪论	1	2.4.1 算术运算符和算术表达式	33
1.1 计算机前沿技术	1	2.4.2 赋值运算符和赋值表达式	34
1.1.1 大数据	1	2.4.3 关系运算符和关系表达式	36
1.1.2 云计算与边缘计算	2	2.4.4 逻辑运算符和逻辑表达式	37
1.1.3 人工智能	5	2.4.5 位运算	38
1.1.4 信息检索	6	2.4.6 其他运算	41
1.2 计算机基础	7	2.5 MATLAB 概述	42
1.2.1 计算机系统	7	2.6 MATLAB 语法基础	47
1.2.2 数制转换	9	2.6.1 MATLAB 的数据类型	47
1.2.3 信息的数字化	13	2.6.2 常量	49
1.3 计算思维与算法	18	2.6.3 部分常用运算符	49
1.3.1 计算思维	18	2.6.4 变量及其赋值	50
1.3.2 算法	19	2.7 小结	51
1.4 程序设计语言概述	22	习题 2	53
1.5 小结	23	第 3 章 数据的输入/输出	58
习题 1	24	3.1 字符输入/输出函数	58
第 2 章 C 语言与 MATLAB 基础	25	3.2 字符串输入/输出函数	59
2.1 C 语言概述	25	3.3 格式输入/输出函数	59
2.1.1 C 语言简介	25	3.4 MATLAB 的输入/输出	67
2.1.2 C 语言程序结构	25	3.4.1 输入文本数据	67
2.1.3 C 语言编译系统	27	3.4.2 输出文本数据	68
2.2 C 语言语法基础	27	3.4.3 低级文件输入/输出函数	68
2.2.1 字符集	27	3.4.4 注释与标点	69
2.2.2 标识符	28	3.5 实例拓展	69
2.2.3 关键字	28	3.6 小结	70
2.2.4 常量	28	习题 3	71
2.2.5 变量	29	第 4 章 选择结构程序设计	72
2.3 C 语言的数据类型	29	4.1 if 语句	72
2.3.1 整型数据	29		
2.3.2 实型数据	30		
2.3.3 字符型数据	31		
2.4 C 语言运算符与表达式	32		

4.1.1	单分支 if 语句	72	6.3.2	函数调用的方式	115
4.1.2	双分支 if···else 语句	72	6.4	函数的参数	116
4.1.3	多分支选择语句	73	6.5	函数的嵌套调用和递归调用	118
4.1.4	if 语句的嵌套	74	6.5.1	函数的嵌套调用	118
4.2	switch 语句	75	6.5.2	函数的递归调用	119
4.2.1	switch 语句简介	75	6.6	变量的存储类型	125
4.2.2	break 语句在 switch 中的应用	76	6.6.1	变量的作用域与生存期	125
4.3	综合实例	77	6.6.2	变量的存储类型	125
4.4	MATLAB 选择结构	78	6.6.3	局部变量	126
4.4.1	if···else···elseif 结构	78	6.6.4	全局变量和静态全局变量	129
4.4.2	switch···case 结构	79	6.7	内部函数和外部函数	131
4.4.3	try···catch 结构	80	6.7.1	内部函数	132
4.5	实例拓展	81	6.7.2	外部函数	132
4.6	小结	83	6.8	编译预处理命令	133
	习题 4	83	6.8.1	宏替换	133
第 5 章	循环结构程序设计	89	6.8.2	文件包含	136
5.1	while 语句	89	6.8.3	条件编译	137
5.2	do···while 语句	90	6.9	综合实例	140
5.3	for 语句	91	6.10	MATLAB 函数简介	142
5.4	循环结构的嵌套	93	6.10.1	m 文件概述	142
5.5	break 语句和 continue 语句在循环结构中的应用	94	6.10.2	用 m 文件实现 MATLAB 函数	143
5.6	综合实例	95	6.11	实例拓展	144
5.7	MATLAB 循环结构	97	6.12	小结	146
5.7.1	while 语句	97		习题 6	147
5.7.2	for 语句	98	第 7 章	数组	156
5.8	实例拓展	99	7.1	一维数组	156
5.9	小结	101	7.1.1	一维数组的定义	156
	习题 5	101	7.1.2	一维数组元素的引用	157
第 6 章	函数与编译预处理	108	7.1.3	一维数组的初始化	157
6.1	函数概述	108	7.2	二维数组	160
6.2	函数的定义和说明	109	7.2.1	二维数组的定义和引用	160
6.2.1	函数的定义	109	7.2.2	二维数组元素的初始化	162
6.2.2	函数的返回值	111	7.3	多维数组	163
6.2.3	函数的说明	113	7.3.1	多维数组的定义、使用与存储	163
6.3	函数的调用	114	7.3.2	多维数组的初始化	164
6.3.1	函数调用的一般形式	114			

7.4 字符串与字符数组·····	165	8.6 指针与函数·····	216
7.4.1 字符串与字符数组的 概念·····	165	8.6.1 指针作为函数参数·····	216
7.4.2 字符数组的初始化·····	166	8.6.2 函数指针变量·····	225
7.4.3 字符数组的输入/输出·····	167	8.6.3 指针型函数·····	227
7.4.4 字符串处理函数·····	170	8.7 指针数组与 main()函数的 参数·····	229
7.5 字符串数组·····	172	8.8 综合实例·····	233
7.6 数组作为函数参数·····	173	8.9 实例拓展·····	235
7.6.1 数组元素作为函数参数·····	173	8.10 小结·····	240
7.6.2 地址量作为函数参数·····	173	习题 8·····	241
7.7 综合实例·····	178	第 9 章 构造数据类型·····	251
7.8 MATLAB 数组·····	179	9.1 结构体·····	251
7.8.1 向量的创建·····	180	9.1.1 结构体的定义·····	251
7.8.2 矩阵的创建·····	180	9.1.2 结构体变量的说明·····	252
7.8.3 矩阵元素的提取与替换·····	181	9.1.3 结构体变量的引用·····	254
7.8.4 矩阵元素的重排和 复制排列·····	183	9.1.4 结构体数组·····	255
7.8.5 矩阵的翻转和旋转·····	183	9.1.5 指向结构体的指针·····	257
7.8.6 矩阵的生成与提取函数·····	184	9.1.6 结构体与函数·····	259
7.8.7 应用实例·····	184	9.1.7 动态存储分配·····	261
7.9 实例拓展·····	184	9.1.8 结构体与链表·····	263
7.10 小结·····	188	9.2 共用体·····	267
习题 7·····	189	9.3 枚举·····	270
第 8 章 指针·····	197	9.4 自定义数据类型·····	271
8.1 指针的基本概念·····	197	9.5 综合实例·····	272
8.2 指针变量的声明与使用·····	198	9.6 MATLAB 的结构体数据类型·····	275
8.2.1 指针变量的声明·····	198	9.6.1 结构体数组的创建·····	275
8.2.2 指针变量的赋值与使用·····	198	9.6.2 结构体数组的操作·····	280
8.2.3 二级指针·····	201	9.7 小结·····	280
8.3 指针运算·····	202	习题 9·····	281
8.3.1 赋值运算·····	202	第 10 章 文件·····	286
8.3.2 算术运算·····	202	10.1 文件的概念·····	286
8.3.3 关系运算·····	204	10.2 文件指针·····	287
8.4 指针与数组·····	204	10.3 文件的打开与关闭·····	287
8.4.1 指针与一维数组·····	204	10.3.1 文件打开函数·····	287
8.4.2 指针与二维数组·····	208	10.3.2 文件关闭函数·····	289
8.5 指针与字符串·····	211	10.4 文件的读/写·····	290
8.5.1 指向字符串的指针·····	211	10.4.1 字符读/写函数 fgetc()和 fputc()·····	290
8.5.2 字符指针与字符数组的 比较·····	214		

10.4.2	字符串读/写函数 fgetc()和 fputc().....	293	11.1.4	多态.....	310
10.4.3	数据块读/写函数 fread()和 fwrite().....	294	11.2	C++概述.....	311
10.4.4	格式化读/写函数 fscanf()和 fprintf().....	296	11.3	C++面向对象的特性.....	311
10.5	文件的随机读/写.....	297	11.4	C++的词法与规则.....	311
10.5.1	文件定位.....	297	11.5	C++程序结构的组成.....	312
10.5.2	文件的随机读/写函数.....	297	11.6	C++程序的开发步骤.....	312
10.6	文件检测函数.....	298	11.7	C++程序示例.....	312
10.7	综合实例.....	299	11.8	面向对象的程序设计方法.....	314
10.8	MATLAB 文件操作.....	302	11.8.1	结构化程序设计.....	314
10.8.1	文件的打开与关闭.....	302	11.8.2	面向对象程序设计.....	314
10.8.2	二进制文件的读/ 写操作.....	303	11.8.3	结构化方法与面向 对象方法的比较.....	314
10.8.3	文本文件的读/写操作.....	304	11.9	小结.....	315
10.8.4	MATLAB 的字符串 操作.....	305	习题 11	315
10.9	小结.....	305	第 12 章	C/C++与 MATLAB 混合编程.....	317
习题 10	306	12.1	软件开发.....	317
第 11 章	C++面向对象程序设计基础.....	310	12.2	混合编程概念.....	318
11.1	面向对象的基本概念.....	310	12.3	混合编程开发实例.....	319
11.1.1	类和对象.....	310	12.4	小结.....	327
11.1.2	封装.....	310	附录 A	C 语言常用库函数.....	328
11.1.3	继承.....	310	附录 B	MATLAB 函数表.....	333
			参考文献	352

第1章 绪 论

1.1 计算机前沿技术

1.1.1 大数据

IT行业的又一次技术变革，大数据的浪潮汹涌而至，对国家治理、企业决策和个人生活产生深远影响，并将成为继云计算、物联网之后信息技术产业领域又一重大创新变革。未来的十年将是一个“大数据”引领的智慧科技时代。随着社交网络的逐渐成熟，移动带宽的升级，云计算、物联网应用更加丰富，更多的传感设备、移动终端接入网络，由此而产生的数据将比历史上的任何时期都要多，增长速度也将比历史上的任何时期都要快。

1. 大数据的概念

大数据 (Big Data) 是一个涵盖多种技术的概念，简单地说，大数据是指无法在一定时间内用常规软件工具对其内容进行抓取、管理和处理的数据集合。

2. 大数据的特征

IBM 将大数据理念定义为 4 个 V，即大量化 (Volume)、多样性 (Variety)、速度快 (Velocity) 及由此产生的价值 (Value)。

(1) 大量化 (Volume)

“大”是指数据规模，大数据体量一般都达到了 PB 量级以上。1PB 等于 1024TB，1TB 等于 1024GB，那么 1PB 等于 1024×1024GB。关于数据容量的计算请读者参考本书 1.2.1 节相关内容。

(2) 多样性 (Variety)

网络中单一用户提交的数据缺乏价值，不能称为大数据。但多用户乃至海量用户数据中 (年龄、学历、爱好、性格等)，每个人的特征都不一样，这也就是大数据的多样性。地域和时间范围越大，多源异构数据的多样性就会越强，每个地区、每个时段，都会存在各种各样的数据，如文本、图片、音频、视频、机器数据 (地理位置信息、网络日志) 等。

(3) 速度快 (Velocity)

对“速度快”的理解可分为两方面：

一方面是数据的产生速度快，物联网、云计算、移动互联网、车联网、手机、平板电脑、PC，以及遍布地球各个角落的各种各样的传感器，无一不是数据的来源或者承载方式；

另一方面是数据的处理速度快，数据处理遵循“1 秒定律” (或秒级定律)，一般要在秒级时间范围内通过算法对各种类型的数据进行处理并给出分析结果，获得高价值的数据信息，这也是“1 秒定律”与传统的数据挖掘技术的本质区别。

(4) 价值 (Value)

大数据的价值体现需要根据业务逻辑,通过强大的机器算法迅速地完成数据的价值“提纯”。假如有 1PB 以上 20~35 岁年轻人的上网数据,通过分析这些数据,可以建立用户兴趣模型,指导产品的设计、营销等。如果有几百万病人的数据,根据这些数据进行分析,就能预测疾病的发生,这些都体现了大数据的价值。

3. 大数据的应用

(1) 落地行业应用

大数据无处不在,其广泛应用于制造、金融、医疗、汽车、广告、电信、能源、餐饮和娱乐等各个领域。例如,在制造领域,可利用工业大数据提升制造业水平,包括实现产品故障诊断与预测、分析工艺流程、改进生产工艺、优化生产过程能耗、工业供应链分析与优化、生产计划与排程等;在金融领域,大数据在高频交易、社交情绪分析和信贷风险分析三大金融创新领域发挥重大作用;在汽车领域,利用大数据和物联网技术设计的无人驾驶汽车,在不远的将来或将走入我们的日常生活;在广告领域,借助大数据技术,可以分析客户行为,进行商品个性化推荐和针对性广告投放。

(2) 提高决策能力

基于大数据的决策信息完整性越来越高,理性决策越来越广泛,决策的技术含量和知识含量大幅提高,实现了很多过去难以想象的重大解决方案。

宏观层面,大数据使经济决策部门可以更敏锐地把握经济走向,制定并实施科学的经济政策;微观方面,大数据可以提高企业经营决策水平和效率,推动创新,给企业、行业带来价值。

(3) 优化资源配置

个人根据大数据可以选择适合自己的工作、舒适的居住条件、便利的出行方式等;企业依据大数据可以选择自己的生产经营方向,精细化分工,为客户量身定做产品,实现利益最大化;政府管理部门依据大数据可以调配公共资源,构建健康发展的和谐社会。

1.1.2 云计算与边缘计算

至今,云计算技术广泛应用,随着 5G 的到来,物联网应用的爆发,边缘计算可解决工业物联网的实际业务困难,受到业界越来越多的关注。

1. 云计算

云计算中的“云”在某些方面具有自然界中云的特征,其体量庞大,可动态伸缩,边界模糊且飘忽不定,不好定位但确实存在。那么,到底什么是云及云计算呢?

(1) 云与云计算的概念

“云”就是资源池,能够存放各种 IT 资源,无论是硬件设备还是软件平台及其应用系统,都能够存放在云中。例如,你有很多东西,家里放不下了,放到一个特定的地方保存,可随时提取,这里的“东西”可指数据、软件、服务等,而“特定的地方”就可指云。“云”也可以理解为能自我维护和管理的虚拟计算资源,通常是一些大型服务器集群,包括计算服务器、存储服务器和宽带资源等。

“云计算”是一种商业计算模型。它将计算任务分布在大量计算机构成的资源池上,使各种应用系统能够根据需要获取计算力、存储空间和信息服务。“云计算”的本质是通过网络按

需提供可动态伸缩的廉价计算服务。

云计算将计算资源集中起来,并通过专门的软件实现自动管理,无须人为参与。用户可以动态申请部分资源,支持各种应用程序的运转,无须为烦琐的细节而烦恼,能够更加专注于自己的业务,有利于提高效率、降低成本和技术创新。

云计算的核心理念是资源池,这与2002年就提出的网格计算池(Computing Pool)的概念非常相似。网格计算池将计算和存储资源虚拟成为一个可以任意组合分配的集合,池的规模可以动态扩展,分配给用户的处理能力可以动态回收重用。这种模式能够大大提高资源的利用率,提升平台的服务质量。

(2) 云存储

云存储是根据云计算延伸出来的新概念,是指通过集群应用、网络技术或分布式文件系统等功能,将网络中各种不同的存储设备通过应用软件集合起来进行协同工作,共同对外提供数据存储和业务访问功能的一个系统。从某种意义上来说,就是一个以数据存储和管理为核心的云计算系统。

云存储是一种网上在线存储的模式,即把数据存放在通常由第三方托管的多台虚拟服务器中,而非专属的服务器中。托管(Hosting)公司运营大型的数据中心,需要数据存储托管的人通过向其购买或租赁存储空间的方式,来满足数据存储的需求。数据中心运营商根据用户的需求,在后端准备存储虚拟化的资源,并将其以存储资源池(Storage Pool)的方式提供。用户通过Web服务应用程序接口(API)或Web用户界面得到云存储服务。

目前的云存储模式主要有两种:一种是文件的大容量分享,有些存储服务提供商(Storage Services Provider, SSP)甚至号称无限容量,用户可以把数据文件保存在云存储空间中。另一种是云同步存储模式,例如,百度云盘、Dropbox、Skydrive、Google的GDrive,以及Apple的iCloud等SSP提供的云同步存储业务。

云存储具备以下三方面的优势:

① 存储管理可以实现自动化和智能化,所有的存储资源被整合到一起,用户看到的是单一存储空间;

② 可提高存储效率,通过虚拟化技术解决存储空间的浪费,可以自动重新分配数据,提高存储空间的利用率,同时具备负载均衡、故障冗余功能;

③ 能够实现规模效应和弹性扩展,降低运营成本,避免资源浪费。

(3) 云服务特点

① 超大规模。“云”具有相当的规模,Google公司的“云”已经拥上百万台服务器;IBM、亚马逊、微软、雅虎、阿里巴巴、百度和腾讯等公司的“云”均拥有几十万台服务器。“云”能赋予用户前所未有的计算能力。

② 虚拟化。云计算支持用户在任意位置,使用各种终端获取服务。所请求的资源来自“云”,而不是固定的有形的实体。应用在“云”中某处运行,但实际上用户无须了解应用运行的具体位置,只需要一台计算机、平板电脑或手机,就可以通过网络服务来获取各种能力超强的服务。

③ 高可靠性。“云”使用了数据多副本容错、计算节点同构可互换等措施来保障服务的高可靠性,使用云计算比使用本地计算机更加可靠。

④ 通用性。云计算不针对特定的应用,在“云”的支撑下可以构造出千变万化的应用,同一片“云”可以同时支撑不同的应用运行。

⑤ 高可伸缩性。“云”的规模可以动态伸缩,满足应用和用户规模增长的需要。

⑥ 按需服务。“云”是一个庞大的资源池，用户按需购买，就像使用自来水、电和煤气那样计费。

⑦ 价格低廉。“云”的特殊容错措施使得其可以采用价格低廉的节点来构成云；“云”的自动化管理使数据中心管理成本大幅降低；“云”的公用性和通用性使资源的利用率大幅提升，“云”设施可以建在电力资源丰富的地区，从而大幅降低能源成本。因此，“云”具有前所未有的性价比。

2. 边缘计算

随着万物互联的泛在化发展，IDC（互联网数据中心）预计，到 2020 年全球将有超过 500 亿的终端与设备联网。为了减轻云计算的压力，也为了物联网应用场景所产生的局部、实时、短周期数据的处理与分析的需要，更好地支撑本地业务的实时智能化决策与执行，边缘计算的热度持续上升。

(1) 边缘计算的概念

边缘计算是指在靠近物或数据源头的一侧，采用网络、计算、存储、应用核心能力为一体的开放平台，就近提供服务。

5G 时代，连接设备数量会大量增加，网络边缘侧会产生庞大的数据量。如果这些数据都由核心管理平台来处理，则在敏捷性、实时性、安全性和隐私性等方面都会出现问题。但采用边缘计算，就可以就近处理海量数据，大量设备可以实现高效协同工作，诸多问题将迎刃而解。

边缘计算产业联盟（ECC）针对边缘计算，定义了如下 4 个领域：

- ① 设备域（感知和控制层）；
- ② 网络域（连接和网络层）；
- ③ 数据域（存储和服务层）；
- ④ 应用域（业务和智能层）。

5G 从架构设计之初就支持边缘计算，业务处理功能依托边缘计算可下沉到基站，并对网络会话管理机制进行详细设计，边缘计算行业规范进入标准阶段。

(2) 边缘计算的特点

① 低成本：边缘计算支持数据本地处理，大流量业务本地卸载可以减轻回传压力，有效降低成本。

② 低时延：移动网络数据传输时延是由空口时延（基站到终端之间的传输时延）和网络侧传输时延（基站到核心网之间的传输时延）组成的，时延的降低依赖于空口性能提升、网络侧传输距离缩短。

③ 大带宽：边缘计算平台为企业提供“私有云”托管服务，充分利用运营商网络云化特性，进一步满足大带宽需求。可通过接入运营商的优质网络，为企业提供优质的网络资源和大带宽服务。

3. 云计算与边缘计算的关系

以自动驾驶为例，未来的计算模式是边缘计算与云计算的结合。边缘侧的自动驾驶专用芯片会感知传感器数据并立刻处理、决策，同时，这些处理之后的数据也会在云端汇聚，进行大数据分析、模型搭建和编辑，同时做大规模的仿真，进行深度分析和机器学习，并对边缘侧设备进行更新和升级，使边缘侧设备更智能。“算法+芯片+云计算”构成了未来自动驾驶的三大核心支点。

再以物联网为例，阿里云发布边缘计算产品 Link Edge，确实通过赋予家庭网关计算能力，实现即便是在断网的状态下，生物识别门锁、机器人等仍可正常运作。但是，如果加上云计算，基于云端的大数据分析和判断，在联动的前提下，整个家庭场景的智能设备将变得更为个性化和智能化，例如，关上门时扫地机器人就开始工作等。

由此可以看出，边缘侧设备在大数据处理、大数据存储、应用程序开发、机器学习和人工智能等方面的处理能力无法与云端相比。同时，云端的应用设计、开发、测试、部署、管理等功能是开发边缘应用的关键。

结合上面的例子可以看出，提供边缘计算能力的设备主要在前端，负责数据的实时采集、计算和处理。但是，大多数的数据并不是一次性数据，那些经过处理的数据需要在系统中进行留存，用于算法训练、数据验证等。这时就需要一个大容量的“容器”，而这个是边缘计算所没有的。在这个“容器”中，这些数据将被存储，用于大数据挖掘、算法训练、用户个性化功能塑造等，这些都是非实时需求，在完成这些操作之后将数据传输给终端设备，从而进一步提升服务质量。这个“容器”就是云计算，云计算做大数据分析挖掘、数据共享，同时进行算法模型的训练和升级，将升级后的算法推送到前端，使前端设备更新和升级，完成自主学习闭环。同时，这些数据也有备份的需要，当边缘计算过程中出现意外情况，存储在云端的数据也不会丢失。

从实际应用看，边缘计算并不能代替云计算，也离不开云计算。未来，云计算将与边缘计算形成一种互补、协同的关系，边缘计算需要与云计算通过紧密协同才能更好地满足各种应用场景的需求。边缘计算将主要负责实时、短周期数据的处理，以及本地业务的实时处理与执行，为云端提供高价值的数据；云计算通过大数据分析，负责非实时、长周期数据的处理，优化输出的业务规则或模型，下放到边缘侧，使边缘计算更好地满足本地的需求，同时完成应用的全生命周期管理。

1.1.3 人工智能

人工智能（Artificial Intelligence, AI）是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门技术科学。

在计算机出现之前人们就幻想着有一种机器可以实现人类的思维，可以帮助人们解决问题，甚至比人类拥有更高的智力。人工智能是计算机科学的一个研究分支，是多年来计算机科学研究发展的结晶。人工智能发展到今天所涉及的学科有：哲学、认知科学、数学、神经生理学、心理学、计算机科学、信息论、控制论、不定性论等。研究的问题有：自然语言处理、知识表现、智能搜索、推理、规划、机器学习、知识获取、组合调度、感知、模式识别、逻辑程序设计软计算、不精确和不确定管理、人工生命、神经网络、复杂系统等。人工智能在以下领域应用广泛。

游戏：人工智能在国际象棋、扑克、围棋等计算机游戏中起着至关重要的作用，机器可以根据启发式知识来思考大量可能的位置并计算出最优的落子位置。

自然语言处理：人工智能可以实现人与理解人类自然语言的计算机之间的交互。如常见的机器翻译系统、人机对话系统等。大数据的产生为人工智能的发展提供了契机。如今，Google 翻译的水平已达到专家级。

大数据、云计算、边缘计算、人工智能之间相互影响、联系紧密，它们的应用将深刻影响着我们的生活和工作。

1.1.4 信息检索

所谓信息检索,就是根据用户需求与一定算法,运用特定策略,从互联网检索出指定信息并反馈给用户的一门检索技术。信息检索依托网络爬虫、检索排序、网页处理、大数据处理、自然语言处理等多种技术,为信息检索用户提供快速、高相关性的信息服务。信息检索技术的核心模块一般包括爬虫、索引、检索和排序等,同时可添加其他辅助模块。

1. 信息检索方式

不同信息检索系统(如搜索引擎)所使用的检索方式不同,检索效率也不同。检索方式大致可分为 4 种,包括全文搜索引擎、元搜索引擎、垂直搜索引擎和目录搜索引擎。它们各有特点,并适用于不同的搜索环境。

① 全文搜索引擎是利用爬虫程序抓取互联网上所有相关文章予以索引的检索方式。一般网络用户适用于全文搜索引擎。这种检索方式方便、简捷,并容易获得所有相关信息。但检索到的信息过于庞杂,因此用户需要逐一浏览并选择所需信息。尤其是在用户没有明确检索意图的情况下,这种检索方式非常有效。

② 元搜索引擎是基于多个搜索引擎结果并对其进行整合处理的二次检索方式,适用于广泛、准确地收集信息。不同的全文搜索引擎由于其性能和信息反馈能力存在差异,导致其各有利弊。元搜索引擎的出现恰恰解决了这个问题,有利于各全文搜索引擎间的优势互补,而且也有利于对全文检索方式进行全局控制,引导全文搜索引擎的持续改善。

③ 垂直搜索引擎是对某一特定行业内数据进行快速检索的一种专业检索方式,适用于有明确检索意图情况下的检索。例如,用户购买机票、火车票、汽车票时,或想要浏览网络视频资源时,都可以直接选用行业内专用搜索引擎,准确、迅速地获得相关信息。

④ 目录搜索引擎是依赖人工收集处理数据并置于分类目录链接下的检索方式,这是网站内部常用的检索方式。该检索方式旨在对网站内信息进行整合处理并分目录呈现给用户,其缺点在于用户需预先了解本网站的内容,并熟悉其主要模块构成。总之,目录搜索方式的适应范围非常有限,且需要较高的人工成本来支持维护。

2. 信息检索应注意的问题

随着互联网和各类智能终端的普及,信息检索逐步成为用户获取网络资源的首选方式之一。然而,目前信息检索面临以下几个问题。

① 网页时效性问题。互联网上的用户众多,数据信息来源极广,互联网上的网页是呈实时动态变化的,网页的更新、删除等变动极为频繁,有时会出现新更新的网页在爬虫程序还来不及抓取就已经被删除的情况,这将大大影响检索结果的准确性。

② 大数据存储问题。爬虫抓取的数据在经过预处理后数据量依然相当庞大,这给大数据存储技术带来相当大的挑战。当前大部分搜索引擎都是利用结构化的数据库来存储数据的,结构化的数据库存储的数据具有高共享、低冗余等特点,然而由于结构化的数据库难以并发查询,因此存在查询效率受限的问题。

③ 检索结果可靠性问题。目前由于数据挖掘技术及计算机硬件的限制使得数据处理准确度未能达到理想程度,而且有一些个人或公司会利用搜索引擎的漏洞,通过作弊手段来干扰检索结果,导致检索结果的可靠性较低。

1.2 计算机基础

1.2.1 计算机系统

一个完整的计算机系统包括计算机硬件系统和软件系统两部分。硬件是计算机系统中看得见、摸得着的物理设备，是计算机工作的物质基础。软件是在计算机中执行某种操作任务的程序及有关文档的集合，是计算机的灵魂，没有安装软件的计算机称为裸机。计算机系统组成如图 1.1 所示。

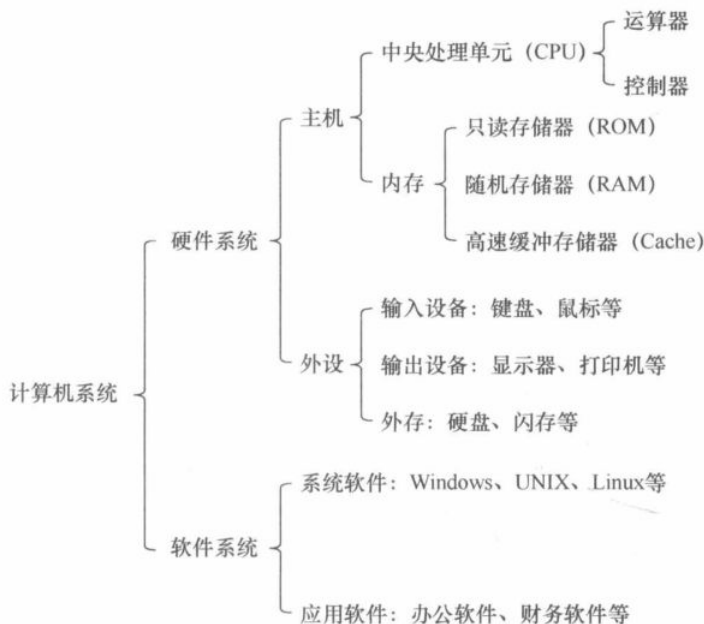


图 1.1 计算机系统组成

1. 冯·诺依曼型计算机

20 世纪 40 年代，计算机界的泰斗之一美籍匈牙利数学家冯·诺依曼针对当时出现的众多能进行计算的机器，对计算机进行定义。

冯·诺依曼提出计算机应该是一个依靠“存储程序”而实现自动工作的机器。根据这一设想，他提出了计算机的 4 项重要的设计思想。

① 计算机应由 5 个基本部分组成：运算器、控制器、存储器、输入设备和输出设备。

② 采用存储程序的方式，程序和数据存放在同一个存储器中。

③ 指令在存储器中按执行顺序存放，由指令计数器指明要执行的指令所在的单元地址，一般按顺序递增，但可按运算结果或外界条件而改变。

④ 机器以运算器为中心，输入/输出设备与存储器间的数据传送都通过运算器。

这就是著名的冯·诺依曼原理。根据这一原理，要组成一台计算机实际上需要 5 个部分的内容，也称为计算机的 5 个部件，如图 1.2 所示。

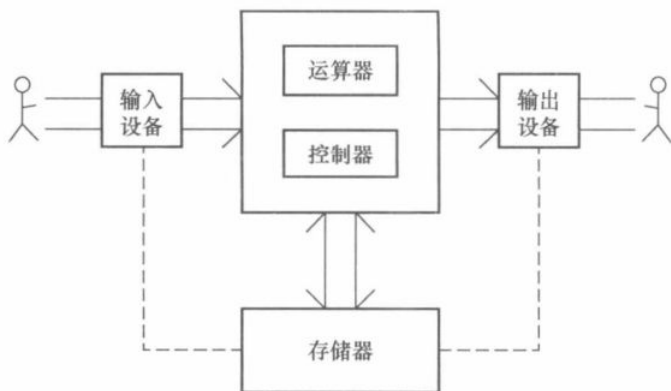


图 1.2 计算机的 5 个部件

2. 存储程序的概念

依照冯·诺依曼原理，计算机 5 个部件可以看成相互独立又紧密连接在一起的整体，连接是为了实现信息交流。在这些信息中，按照信息特征可以分为三种：数据信息、控制信息和地址信息。数据信息用来表示二进制数，控制信息就是计算机的操作指令，而地址信息就是信息存储的存储单元的编码。为了方便设计，计算机中的三种信息都有各自的信息传输专线，这些传输专线被称为总线。因此，计算机内部有三大总线：数据总线（DBus）、控制总线（CBus）、地址总线（ABus）。了解了这三大总线，我们就可得到一个更接近实际情况的计算机结构，如图 1.3 所示。

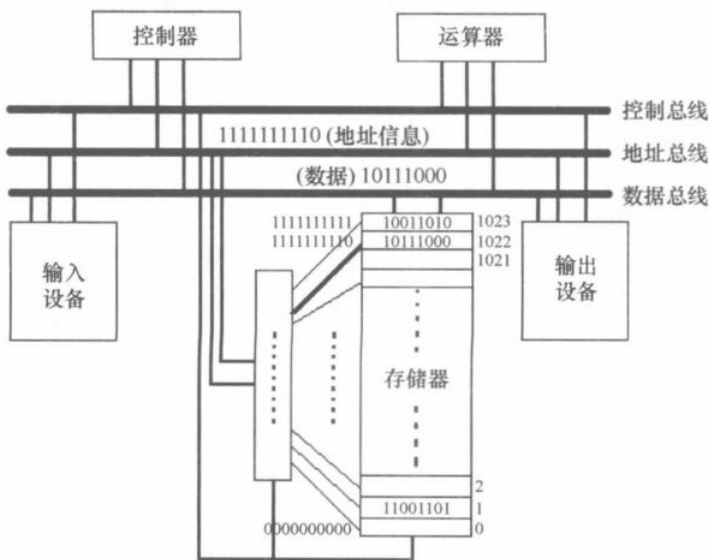


图 1.3 计算机结构

在图 1.3 中，计算机的各个部件通过三大总线紧密联系在一起，存储器的主要功能是对信息进行保存。存储器由大量可以存储二进制数据的存储单元构成，习惯上，我们称 8 个二进制位（一字节）为一个存储单元，每个存储单元都可独立被访问。由于每个单元都要求能被独立操作，因此，需要为每个单元进行编号。如图 1.3 所示，存储器左边的编码“000000000”和“111111111”就是存储单元的编号，它们对应的是 0 号和 1023 号存储单元，这个编号在计算机中称为存储器地址。

由于在计算机中需要存储的二进制数很多、很大，为了规范表示，对二进制数定义了一些数量单位以方便交流。以一个二进制位为基数，称为 1bit（位），具体对应关系如下：