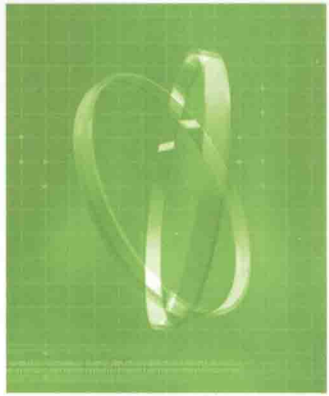

计算机专业英语

主　编　袁　满
副主编　张剑飞　杨冬风

English in Computer Science

哈爾濱工業大學出版社

计算机专业英语

English in Computer Science

主　编　袁　满
副主编　张剑飞　杨冬风

哈爾濱工業大學出版社

内 容 提 要

全书共分 10 章，其中第 1 章为计算机组成；第 2 章为操作系统；第 3 章为数据结构与算法；第 4 章为面向对象软件工程开发基本原理；第 5 章为数据库管理系统；第 6 章为计算机网络与 Internet；第 7 章为网络安全；第 8 章为人工智能；第 9 章为多媒体；第 10 章为分布式系统。本书给出了大量的计算机专业词汇、短语，并对文中较难的句子给出了译文。书中所选取的阅读材料均反映了计算机发展的先进技术，能使学生在学习专业英语的同时，也及时了解 IT 业新技术。

本书可供大学本科或专科计算机及 IT 相关专业的学生使用。

图书在版编目(CIP)数据

计算机专业英语/袁满主编.—哈尔滨：哈尔滨工业大学出版社，2008.7

ISBN 978 - 7 - 5603 - 2744 - 0

Ⅰ.计… Ⅱ.袁 Ⅲ.电子计算机 - 英语 - 高等学校 - 教材 Ⅳ.H31

中国版本图书馆 CIP 数据核字(2008)第 102060 号

责任编辑 王桂芝 贾学斌
出版发行 哈尔滨工业大学出版社
社　　址 哈尔滨市南岗区复华四道街 10 号　邮编 150006
传　　真 0451 - 86414749
网　　址 http://hitpress.hit.edu.cn
印　　刷 哈尔滨工业大学印刷厂
开　　本 880mm×1230mm　1/32　印张 10.5　字数 302 千字
版　　次 2008 年 8 月第 1 版　2008 年 8 月第 1 次印刷
书　　号 ISBN 978 - 7 - 5603 - 2744 - 0
定　　价 20.00 元

前　言

计算机科学与技术是目前发展最快的技术之一,而新技术几乎都是首先以英文为主要形式进行传播的。作为计算机专业的人才只有掌握并精通计算机专业英语,才能把握计算机发展的前沿动态。计算机专业英语是继大学生基础英语之后的一门实用的应用型语言。通过计算机专业英语的系统学习,可以使学生掌握计算机各个专业领域的一些重要词汇、句法及表达方式,为进行英文文献阅读、英文写作及交流等奠定一个坚实的基础。

本书在编写过程中参考了大量的外文文献,本着既要能反映计算机发展的最新技术,又要能够让学生产生兴趣,并且尽量不是太抽象与难理解这样的原则进行编写的。所以总结起来,本教材具有如下一些特点:

第一,尽可能多地覆盖计算机专业的各个领域,内容丰富;

第二,将经典技术与最新技术相结合,突出新技术的先进性与广泛的应用性;

第三,对文中难懂的词汇、短语和句子,均以黑体或下划线标出,并在该单元后给出译文,方便学生阅读与教师参考。

本书涉及内容比较广泛,涵盖了计算机科学技术的多个领域,既可作为高等院校计算机各专业本、专科学生的专业英语教材,也可以作为计算机专业研究人员的英语学习参考资料。

本书由大庆石油学院袁满主编，负责第9章和第10章的编写；齐齐哈尔大学张剑飞与黑龙江八一农垦大学杨冬风副主编，其中张剑飞负责第1、2、5、6章的编写，杨冬风负责第3、4、7、8章的编写。参加编写的还有齐齐哈尔大学的潘海珠、大庆石油学院的王梅、郭玲玲和赵建民几位老师，在这里对几位参编的老师深表谢意。

由于作者水平有限，难免有疏漏和不妥之处，恳请读者提出宝贵意见，以便不断完善。

编　者

2008年5月

CONTENTS

1 Principles of Computer Organization

Introduction ······ 1
1.1 Computer Hardware ······ 1
1.2 Processor ······ 5
1.3 Memory Systems ······ 8
1.4 Input/Output System ······ 14
1.5 Buses and Controllers ······ 21
1.6 USB—Universal Serial Bus ······ 28

2 Operating System

Introduction ······ 32
2.1 Summary of Operating System ······ 33
2.2 Operating System Concepts ······ 36
2.3 Case Studies ······ 44
2.4 Operating System Present and Future ······ 48

3 Data Structures and Algorithm

Introduction ······ 51
3.1 Abstract Data Types ······ 53
3.2 List ······ 57
3.3 Graphs ······ 63
3.4 Binary Tree Structure ······ 68
3.5 Algorithm Analysis ······ 72

4 Principles of Object-oriented Software Development

Introduction 80
4.1 Terminology 81
4.2 Paradigms of Programming 88
4.3 The object-oriented Software Life-cycle 92
4.4 Unified Modeling Language 100

5 Database Management System

Introduction 106
5.1 What Is a Relational Database 107
5.2 Introduction to a Database Management System (DBMS) 108
5.3 Database Objects of SQL Server 114
5.4 Client/Server Architecture and Browser/Server Technology 119
5.5 New Forms of Databases and Data Warehouses 124
5.6 SQL Server 2000 Product Overview 127
5.7 Distributed Databases 135
5.8 The Technology of Middleware 142
5.9 XML 149

6 Computer Network and Internet

Introduction 170
6.1 Telecommunications and Information Superhighway 171
6.2 Internet and Its History 177
6.3 Computer Network 184
6.4 Open System Interconnection Reference Model 190
6.5 Internet Protocol Stack 200
6.6 IPV6 and Mobile IPV6 210

7 **Network security**

7.1 The OSI Security Architecture ········ 216
7.2 IP Security ········ 221
7.3 Web Security ········ 226
7.4 Viruses and Related Threats ········ 231
7.5 Firewalls ········ 237

8 **Artificial Intelligence**

Introduction ········ 245
8.1 Overview of Artificial Intelligence ········ 246
8.2 Lisp ········ 250
8.3 Learning ········ 255
8.4 Image Understanding ········ 261

9 **Multimedia**

9.1 Introduction to Multimedia ········ 267
9.2 Multimedia Authoring and Tools ········ 271
9.3 Multimedia Data Compression ········ 273
9.4 Multimedia Network Communications and Applications ········ 284

10 **Distributed System**

10.1 What Is a Distributed System ········ 293
10.2 Goal ········ 295
10.3 Client/Server ········ 305
10.4 Grid Computing ········ 313

Bibliography ········ 327

1

Principles of Computer Organization

Introduction

The generic computer contains input devices, a computational unit, and output devices. The computational unit is the computer's heart, and usually consists of a **central processing unit** (**CPU**), a memory, an input/output (I/O) interface and other important components. To get a better understanding of computer organization, a brief overview of the computers organization is given in the flowing.

1.1 Computer Hardware

We build **computer** to solve problems. Early computer solved mathematical and engineering problems, and later computers emphasized **information processing** for business applications, see Fig. 1.1. Today, computers also control machines as diverse as automobile engines, robots, and microwave ovens. A computer system solves a problem from any of these domains by accepting input, processing it, and producing output.

Computer systems consist of **hardware** and **software**. Hardware is the physical part of the system. Once designed, hardware is difficult and

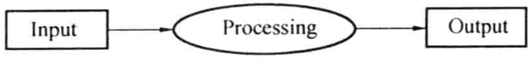

Fig. 1.1 The three activities of a computer system

expensive to change. Software is the set of **programs** that instruct the hardware and is easier to modify than hardware. Computers are valuable because they are general-purpose machines that can solve many different kinds of problems, as opposed to special-purpose machines that can each solve only one kind of problem[1]. Different problems can be solved with the same hardware by supplying the system with a different set of **instructions**, that is, with different software.

Every computer has four basic hardware components.

- **Input devices.**
- **Output devices.**
- **Main memory.**
- Central processing unit (CPU).

Fig. 1.2 shows these components in a block diagram. The lines between the blocks represent the flow of information flows from one component to another on the **bus**, which is simply a group of wires connecting the components. Processing occurs in the CPU and main memory. The organization in Fig. 1.2, with the components connected to each other by the bus, is common. However, other configurations are possible as well.

Computer hardware is often classified by its relative physical size.

- Small **microcomputer**;
- Medium **minicomputer**;
- Large **mainframe**.

Just the CPU of a mainframe often occupies an entire cabinet. Its input/output (I/O) devices and memory might fill an entire room. Microcomputers can be small enough to fit on a desk or in a briefcase. As

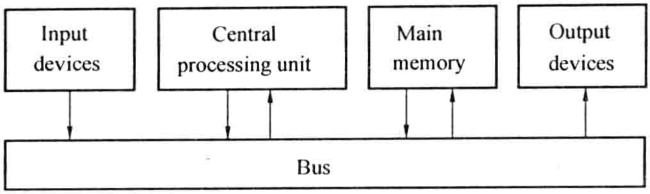

Fig.1.2 Block diagram of the four components of a computer system

technology advances, the amount of processing previously possible only on large machines becomes possible on smaller machines. Microcomputers now can do much of the work that only minicomputers or mainframes could do in the past.

The classification just described is based on physical size as opposed to storage size. A computer system user is generally more concerned with storage size, because that is a more direct indication of the amount of useful work that the hardware can perform[2].

Speed of computation is another characteristic that is important to the user. Generally speaking, users want a fast CPU and large amounts of storage, but a physically small machine for the I/O devices and main memory.

When computer scientists study problems, therefore, they are concerned with space and time, the space necessary inside a computer system to store a problem and the time required to solve it. They commonly use the metric prefixes of Table 1.1 to express large or small quantities of space or time.

Table 1.1 Prefixes for powers of 10

Multiple	Prefix	Abbrev
10^9	giga-	G
10^6	mega-	M
10^3	kilo-	k
10^{-3}	milli-	m
10^{-6}	micro-	μ
10^{-9}	nano-	n

Example Suppose it takes 4.5 microseconds, also written 4.5 μs, to transfer some information across the bus from one component to another. (a) How many seconds are required for the transfer? (b) How many transfers can take place during one minute?

(a) A time of 4.5 μs is 4.5×10^{-6} from Table 1.1 or 0. 000 004 5 s. (b) Because there are 60 seconds in one minute, the number of times the transfer can occur is (60 s)/(0. 000 004 5 s/transfer) or 13 300 000 transfers. Note that since the original value was given with two significant figures, the result should not be given to more than two or three significant figures.

Table 1.1 shows that in the metric system the prefix kilo- is 1 000 and mega- is 1 000 000. But in computer science, a kilo- is 2^{10} or 1 024. The different between 1 000 and 1 024 is less than 3%, so you can think of a computer science kilo- as being about 1 000 even though it is a little more. The same applies to mega- and giga-, as in Table 1.2. This time, the approximation is a little worse, but for mega-, it is still within 5%.

Table 1.2 Computer science values of the large prefixes

Prefix	Computer science value
giga-	2^{30} = 1 073 741 824
mega-	2^{20} = 1 048 576
kilo-	2^{10} = 1 024

Vocabulary

1. computer	[kəm'pju:tə]	*n*. 计算机
2. hardware	['ha:dwεə]	*n*. 硬件
3. software	['sɔftwεə]	*n*. 软件
4. program	['prəugræm]	*n*. 程序
5. instruction	[in'strʌkʃən]	*n*. 指令

6. bus [bʌs] n. 总线
7. microcomputer ['maikrəukəmpju:tə(r)] n. 微型计算机
8. minicomputer ['minikəmˌpju:tə] n. 小型计算机
9. mainframe ['meinfreim] n. 主机,大型机

Phrases

1. central processing unit (CPU) 中央处理器
2. information processing 信息处理
3. input device 输入设备
4. output device 输出设备
5. main memory 主存储器
6. general-purpose machine 通用计算机
7. special-purpose machine 专用计算机

Notes

[1] Computers are valuable because they are general-purpose machines that can solve many different kinds of problems, as opposed to special-purpose machines that can each solve only one kind of problem.

由于计算机是能够解决多种不同类型问题的通用机器,而不是仅解决某一类问题的专用机器,所以它们具有(重要)价值。

[2] A computer system user is generally more concerned with storage size, because that is a more direct indication of the amount of useful work that the hardware can perform.

计算机用户通常更关心存储容量的大小,因为它是硬件能够执行的有用工作量的更直接的指示。

1.2 Processor

A **processor** is a functional unit that interprets and carries out instructions. Every processor comes with a unique set of operations such as ADD, STORE, or LOAD that represent the processor's instruction set[1]. Computer designers are fond of calling their computers machines, so the instruction set is sometimes referred to as machine instructions and the binary language in which they are written is called **machine language**! You shouldn't confuse the processor's instruction set with the instructions found in high-level programming languages, such as Basic or Pascal.

An instruction is made up of **operations** that specify the function to be performed and **operands** that represent the data to be operated on. For example, if an instruction is to perform the operation of adding two numbers, it must know (1) what the two numbers are and (2) where the two numbers are. When the numbers are stored in the computer's memory, they have their addresses to indicate where they are. So if an operand refers to data in the computer's memory it is called an address. The processor's job is to retrieve instructions and operands from memory and to perform each operation[2]. Having done that, it signals memory to send it the next instruction.

This step-by-step operation is repeated over and over again at awesome speed. A timer called a **clock** releases precisely timed electrical signals that provide a regular pulse for the processor's work. The term that is used to measure the computer's speed is borrowed from the domain of electrical engineering and is called a **megahertz** (MHz), which means million cycles per second. For example, in an 8(MHz) processor, the computer's clock ticks 8 million times to every 1 second tick of an ordinary clock.

A processor is composed of two functional units: a **control unit** and an arithmetic/logic unit—and a set of special workspaces called **registers**.

1. The Control Unit

The control unit is the functional unit that is responsible for supervising the operation of the entire computer system. In some ways, it is analogous to a telephone switch-board with intelligence because it makes the connections between various functional units of the computer system and calls into operation each unit that is required by the program currently in operation.

The control unit fetches instructions from memory and determines their types or decodes them. It then breaks each instruction into a series of simple small steps or actions. By doing this, it controls the step-by-step operation of the entire computer system.

2. The Arithmetic and Logic Unit

The **arithmetic and logic unit** (**ALU**) is the functional unit that provides the computer with logical and computational capabilities. Data are brought into the ALU by the control unit, and the ALU performs whatever arithmetic or logic operations are required to help carry out the instruction.

Arithmetic operations include adding, subtracting, multiplying, and dividing. Logic operations make a comparison and take action based on the results. For example, two numbers might be compared to determine if they are equal. If they are equal, processing will continue; if they are not equal, processing will stop.

3. Registers

A register is a storage location inside the processor. Registers in the control unit are used to keep track of the overall status of the program that is running. Control unit registers store information such as the current instruction, the location of the next instruction to be executed, and the operands of the instruction. In the ALU, registers store data items that are added, subtracted, multiplied, divided, and compared. Other registers store the results of arithmetic and logic operations.

An important factor that affects the speed and performance of a processor is the size of the registers. Technically, the term **word size** (also called word length) describes the size of an operand register, but it is also used more loosely to describe the size of the pathways to and from the processor. Currently, word sizes in general purpose computers range from 8 to 64 bits. If the operand registers of a processor are 16 bits wide, the processor is said to be a 16 – bit processor.

Vocabulary

1. processor	[ˈprəusesə]	*n*. 处理器
2. operation	[ɔpəˈreiʃən]	*n*. 操作,操作指令
3. operand	[ˈɔpəˌrænd]	*n*. 操作数
4. clock	[klɔk]	*n*. 时钟
5. megahertz(MHz)	[ˈmegəˌhə:ts]	*n*. 兆赫
6. register	[ˈredʒistə]	*n*. 寄存器

Phrases

1. machine language　机器语言
2. control unit　控制部件
3. arithmetic and logic unit (ALU)　算术/逻辑部件
4. word size　字长

Notes

[1] Every processor comes with a unique set of operations such as ADD, STORE, or LOAD that represent the processor's instruction set.

每个处理器都具有如加、存储或装入这样代表处理器指令集的一组独特的操作集。

[2] The processor's job is to retrieve instructions and operands from memory and to perform each operation.

处理器的工作是从内存检索指令和操作数并完成每一个操作。

1.3 Memory Systems

1. Memory System Desiderata

The memory system has three desiderata.

(1) Size: Infinitely large, no constraints on program or data set size.

(2) Speed: Infinitely fast, latency equal to the fastest memory technology available.

(3) Cost: The per bit cost should approach the lowest-cost technology available.

Clearly these specifications cannot all be achieved as they are mutually exclusive[1]. However, with the semiconductor and magnetic memory technology of today, these specifications are closely approximated.

2. Hierarchical Memory

In this section it is shown how designers implement a practical memory that approaches the performance of an ideal memory at reasonable cost. This memory system has a hierarchy of levels: The memory closest to the processor is fast and relatively small, but has a high cost per bit. This level is called the **cache**; The real memory, sometimes known as **main memory**, is slower, larger, and has a lower cost per hit than the cache; The lowest level in the hierarchy is usually a **magnetic disk** that has the longest **latency** and the lowest bandwidth; however, it can be very large and has a very low cost per bit. This hierarchy is illustrated in Fig. 1.3.

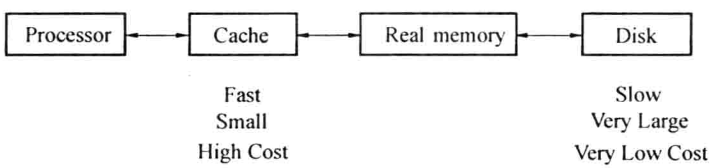

Fig. 1.3 Hierarchical memory

Note that Fig. 1.3 does not include the processor **register file** in the memory hierarchy. The register file is a program-managed cache and is generally not included in the memory system. Also, there can be more