

计算机系列教材

# 计算机专业英语

## (第2版)

邱晓红 吴沧海 主 编  
刘秋明 杨 俊 副主编

清华大学出版社



计算机系列教材

邱晓红 吴沧海 主 编  
刘秋明 杨 俊 副主编

# 计算机专业英语

## (第2版)

贵州师范学院内部使用

清华大学出版社  
北京

## 内 容 简 介

“计算机专业英语”是计算机及相关专业的一门专业基础课程。由于计算机相关核心技术大部分源于英语国家,而且其更新的速度越来越快,如果不掌握计算机专业英语,势必受到语言的制约,严重影响对新技术的理解和消化。本书基于 CDIO 的工程教育理念,结合需要掌握的专业知识点,选取了计算机基础、计算机网络、云计算、数据库、程序设计语言、MSDN、编译原理、离散数学、软件工程、嵌入式系统、数据结构与算法、操作系统等相关内容,增强了计算机相关领域的知识点和英汉专业术语的对应关系。素材取自国内外最近几年计算机科学各个领域的最新教材、专著、论文和网络信息。本书每个单元的内容相对独立,包括课文 A、课文 B、单词、难句分析、语法、技能训练和阅读材料,对较偏、较难的单词进行了注释。修订版增加了人工智能、科技论文写作、参考译文等内容。

本书可以作为教学研究型高等院校和教学(应用)型大学计算机相关专业的专业英语教材,也可供计算机专业人员及其他有兴趣的读者学习参考。

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。侵权举报电话:010-62782989 13701121933

### 图书在版编目(CIP)数据

计算机专业英语/邱晓红,吴沧海主编. —2 版. —北京:清华大学出版社,2019  
(计算机系列教材)

ISBN 978-7-302-52399-4

I. ①计… II. ①邱… ②吴… III. ①电子计算机-英语-教材 IV. ①TP3

中国版本图书馆 CIP 数据核字(2019)第 041433 号

责任编辑:白立军 常建丽

封面设计:常雪影

责任校对:焦丽丽

责任印制:杨 艳

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

邮 编:100084

社总机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:17.25

字 数:398千字

版 次:2011年9月第1版 2019年8月第2版

印 次:2019年8月第1次印刷

定 价:45.00元

产品编号:080914-01

# 前 言

计算机相关核心技术大部分源于英语国家,软件开发工具的主流及程序语言基本要素是英语。信息技术日新月异地发展,如果不掌握计算机专业英语,势必受到语言的制约,严重影响对新技术的理解和消化。所以,从事 IT 行业的人员都需要掌握计算机英语的常用术语和缩略语、常见的语法和惯用法,能借助字典熟练阅读英文文档和技术资料。“计算机专业英语”是计算机及相关专业的一门专业基础课程。

学习计算机专业英语,最好的教学方法是对计算机相关专业课采用双语教学,但教学研究型和教学(应用)型大学的学生,英语的听、说、写能力相对偏弱,双语教学对专业知识的掌握效果不那么理想。所以,为了便于学生掌握计算机专业英语,只能退而求其次,通过选择有趣、较易懂的专业素材或软件工具相关实用内容,让学生重温专业课程的知识点,并学会相关的中英文专业词汇。专业英语教学课程成为各门专业课程知识点融会的平台和载体,体现了 CDIO(构思(Conceive)、设计(Design)、实现(Implement)和运作(Operate))的工程教学理念,摆脱了双语教学难以掌握专业知识的尴尬,也增强了学习专业英语的趣味和实用性,进一步增强了计算机相关领域的应用知识点和英汉专业术语的对应关系。

本书针对教学研究型和教学(应用)型大学的学生英语素质特点,结合需要掌握的专业知识点,从国内外最近几年计算机相关领域的最新教材、专著、论文和网络信息中选取了计算机基础、计算机网络、云计算、数据库、程序设计语言、MSDN、编译原理、离散数学、软件工程、嵌入式系统、数据结构与算法、操作系统等内容素材。全书共分为 11 个单元,每个单元的内容相对独立,包括课文 A、课文 B、单词、难句分析、语法、技能训练和阅读材料,对较偏、难、专业的词汇进行了注释。修订版增加了人工智能、科技论文写作、参考译文等内容。授课教师可根据课时要求及专业特点加以裁减。

邱晓红构想了全书内容和要求,审查各单元内容并统一了全书风格,吴沧海编写了第 4~10 单元及附录 F,刘秋明编写了第 1~3 单元、第 11 单元及附录 G,杨俊编写了附录相关内容,赖谷丹、杨欣参与了本书的素材收集和整理工作。全书由邱晓红统稿并定稿。

本书内容的工程教学理念构想来自“基于布鲁姆教学理论的计算机专业外语课程教学模式的研究和探讨”“教学与科研有机结合提高教学质量和学生综合素质的研究(2008B2ZB04)”“实施本科生全程导师制教育培养模式的研究——以江西省高校软件学院为例(08YB078)”“软件工程融合传统专业复合型创新人才培养模式研究(JXJG-12-6-

15)”等教改课题的研究成果,在此特别感谢江西省教育厅、江西理工大学等有关单位领导和教学管理部门的支持。在本书的编写过程中,我们还参阅了大量的专业英语书籍、英语网站和有关教学研究论文资料,在此向有关作者表示衷心的感谢!在本书的使用过程中,得到许多老师和同学的指正,清华大学出版社对本书的修订出版工作给予了大力的支持,在此表示衷心的感谢!

书中难免有不妥和错误之处,恳请广大读者及同行专家批评指正,以便于再版修订完善。

编者

2018年12月于江西南昌

# 目 录

<b>Unit 1 Computer Structure</b> .....	1
Text A The Function of Computer .....	1
Words .....	9
Phrases .....	11
Exercises .....	12
批注 .....	13
Text B The Future of Computer Technology .....	14
Words .....	18
Phrases .....	19
批注 .....	19
Associated Reading .....	20
批注 .....	20
<b>Unit 2 Programming Language</b> .....	22
Text A Functions in C .....	22
Words .....	24
Phrases .....	24
Exercises .....	24
批注 .....	25
Text B J2ME Related .....	26
Words .....	29
Phrases .....	29
批注 .....	30
Associated Reading .....	30
<b>Unit 3 Discrete Mathematic</b> .....	34
Text A About Discrete Mathematic .....	34
Words .....	37
Phrases .....	38
Exercises .....	39
批注 .....	40
Text B Tree .....	40
Words .....	44

Phrases .....	44
批注 .....	44
Associated Reading .....	45
<b>Unit 4 Software Engineering</b> .....	47
Text A Software Processes .....	47
Words .....	51
Phrases .....	52
Exercises .....	53
批注 .....	54
Text B Introducing the UML .....	54
Words .....	58
Phrases .....	59
批注 .....	60
Associated Reading .....	60
<b>Unit 5 Database</b> .....	63
Text A MySQL Introduction .....	63
Words .....	65
Phrases .....	65
Exercises .....	66
批注 .....	67
Text B Performing Transactions .....	67
Words .....	69
Phrases .....	70
批注 .....	70
Associated Reading .....	70
<b>Unit 6 Embedded System</b> .....	72
Text A What is an Embedded System? .....	72
Words .....	75
Phrases .....	76
Exercises .....	77
批注 .....	78
Text B Getting to Know the Hardware .....	79
Words .....	82
Phrases .....	83
批注 .....	83

Associated Reading .....	84
<b>Unit 7 Computer Network .....</b>	<b>86</b>
Text A Internet Protocol Suite .....	86
Words .....	89
Phrases .....	90
Exercises .....	91
批注 .....	92
Text B Cloud Computing .....	93
Words .....	96
Phrases .....	97
批注 .....	97
Associated Reading .....	98
<b>Unit 8 Data Structure .....</b>	<b>100</b>
Text A Data Structures and Algorithms in Java .....	100
Words .....	102
Phrases .....	103
Exercises .....	103
批注 .....	104
Text B The Introduction of Two Important Data Structures .....	104
Words .....	106
Phrases .....	107
批注 .....	107
Associated Reading .....	107
批注 .....	109
<b>Unit 9 Microsoft Developer Network .....</b>	<b>110</b>
Text A What is MSDN? .....	110
Words .....	112
Phrases .....	113
Exercises .....	114
批注 .....	115
Text B Getting to Know MSDN .....	115
Words .....	118
Phrases .....	118
批注 .....	119
Associated Reading .....	119

<b>Unit 10 Compilers Principles</b> .....	121
Text A The Science of Code Optimization .....	121
Words .....	123
Phrases .....	124
Exercises .....	124
批注 .....	125
Text B Optimizations for Computer Architectures .....	125
Words .....	128
Phrases .....	128
批注 .....	128
Associated Reading .....	129
<b>Unit 11 Operating System</b> .....	136
Text A Operating System Overview .....	136
Words .....	139
Phrases .....	140
批注 .....	140
Exercises .....	140
Text B BIOS or CMOS Setup .....	141
批注 .....	143
Words and Phrases .....	143
Associated Reading .....	144
批注 .....	148
附录 A 计算机专业英语主要句型及翻译技巧 .....	149
附录 B 计算机专业英语的特点 .....	161
附录 C 常见计算机英语缩写 .....	172
附录 D 计算机常用词汇 .....	195
附录 E 工科学生学习英语的基本要求 .....	223
附录 F 部分答案 .....	225
附录 G 参考译文 .....	232
参考文献 .....	265

# Unit 1 Computer Structure

## Text A The Function of Computer

### 1. Central processing unit and microprocessor

A general purpose computer has four main **components**<sup>①</sup>: the arithmetic logic unit (ALU), the control unit, the memory, and the input and output devices (collectively termed I/O). These parts **are interconnected by**<sup>②</sup> buses, often **made of**<sup>③</sup> groups of wires.

Inside each of these parts are trillions of small **electrical circuits**<sup>④</sup> which can be turned off or on by means of an electronic switch. Each circuit represents a bit (binary digit) of information so that when the circuit is on it represents a “1”, and when off it represents a “0” (in positive logic representation).<sup>注1</sup> The circuits are arranged in logic gates so that one or more of the circuits may control the state of one or more of the other circuits.

The control unit, ALU, registers, and basic I/O (and often other hardware closely linked with these) are collectively known as a central processing unit (CPU). Early CPUs were composed of many separate components but since the mid-1970s CPUs have typically been constructed on a single **integrated circuit**<sup>⑤</sup> called a microprocessor.<sup>注2</sup>

Since 1985, numerous processors implementing some version of the MIPS architecture have been designed and widely used. MIPS (an acronym for Microprocessor without Interlocked Pipeline Stages) is a reduced instruction set computer (RISC) instruction set architecture (ISA) developed by MIPS Technologies (formerly MIPS Computer Systems). The early MIPS architectures were 32-bit called MIPS32, with 64-bit versions added later. The MIPS32 instruction set was developed along side the MIPS64 Instruction Set which includes 64-bit instructions. The MIPS32 standard included coprocessor 0 control instructions for the first time. Today, the MIPS32 instruction set is the most common MIPS instruction set, compatible with most CPUs. Due to its relative simplicity, the MIPS32 instruction set is also the most common instruction set taught in computer architecture university courses. MIPS32 Add Immediate Instruction is shown in Figure 1-1.

The control unit (often called a control system or central controller) manages the

- 
- ① 元件, 部件
  - ② 由……联系起来
  - ③ 由……组成
  - ④ 电路
  - ⑤ 集成电路

MIPS32 Add Immediate Instruction

001000		00010	0000000101011110
OP Code	Addr 1	Addr 2	Immediate Value

Equivalent mnemonic: `addi &r1, &r2, 350`

Figure 1-1 Diagram showing how a particular MIPS architecture instruction would be decoded by the control system

computer's various components; it reads and interprets (**decodes**)<sup>⑥</sup> the program instructions, **transforming** them **into**<sup>⑦</sup> a series of control signals which activate other parts of the computer.<sup>注3</sup> Control systems in **advanced**<sup>⑧</sup> computers may change the order of some instructions so as to improve **performance**<sup>⑨</sup>.

A key component common to all CPUs is the program counter: a special memory cell (a register) that keeps track of which location in memory the next instruction is to be read from.

The control system's function is as follows—note that this is a simplified description, and some of these steps may be performed **concurrently**<sup>⑩</sup> or in a different order depending on the type of CPU:

Read the code for the next instruction from the cell indicated by the program counter.

Decode the numerical code for the instruction into a set of commands or signals for each of the other systems.

**Increment**<sup>⑪</sup> the program counter so it points to the next instruction.

Read whatever data the instruction requires from cells in memory (or perhaps from an input device). The location of this required data is typically stored within the instruction code.

Provide the necessary data to an ALU or **register**<sup>⑫</sup>.

If the instruction requires an ALU or **specialized**<sup>⑬</sup> hardware to complete, instruct the hardware to perform the requested operation.

**Write** the result from the ALU **back to**<sup>⑭</sup> a memory location or to a register or perhaps an output device.

Since the program counter is (conceptually) just another set of memory cells, it can be changed by calculations done in the ALU. Adding 100 to the program counter would cause

- ⑥ 解码
- ⑦ 转换
- ⑧ 高级
- ⑨ 性能
- ⑩ 同时发生
- ⑪ 增量
- ⑫ 寄存器
- ⑬ 专门的
- ⑭ 把……写回到

the next instruction to be read from a place 100 locations further down the program. Instructions that modify the program counter are often known as “jumps” and allow for loops (instructions that are repeated by the computer) and often conditional instruction execution (both examples of control flow).

It is noticeable that the **sequence**<sup>⑮</sup> of operations that the control unit goes through to process an instruction is in itself like a short computer program—and indeed, in some more complex CPU designs, there is another yet smaller computer called a microsequencer that runs a microcode program that causes all of these events to happen.

## 2. Arithmetic/Logic Unit(ALU)

The ALU is **capable of**<sup>⑯</sup> performing two classes of operations: **arithmetic**<sup>⑰</sup> and logic.

The set of arithmetic operations that a particular ALU supports may be limited to adding and **subtracting**<sup>⑱</sup> or might include multiplying or dividing or **trigonometry functions**<sup>⑲</sup> (sine, cosine, etc.) and square roots.<sup>注4</sup> Some can only operate on whole numbers (integers) whilst others use floating point to represent real numbers—**albeit with**<sup>⑳</sup> limited precision. However, any computer that is capable of performing just the simplest operations can be programmed to **break down**<sup>㉑</sup> the more complex operations **into** simple steps that it can perform.<sup>注5</sup> Therefore, any computer can be programmed to perform any arithmetic operation—although it will take more time to do so if its ALU does not directly support the operation. An ALU may also compare numbers and return **boolean**<sup>㉒</sup> truth values (true or false) depending on whether one is **equal to**<sup>㉓</sup>, **greater than**<sup>㉔</sup> or **less than**<sup>㉕</sup> the other (“is 64 greater than 65?”).

Logic operations involve Boolean logic: AND, OR, XOR and NOT. These can be useful both for creating complicated conditional statements and processing Boolean logic.

**Superscalar**<sup>㉖</sup> computers may contain **multiple**<sup>㉗</sup> ALUs so that they can process several instructions at the same time. Graphics processors and computers with SIMD and

- 
- ⑮ 一连串, 序列
  - ⑯ 能……
  - ⑰ 算术, 计算
  - ⑱ 减法
  - ⑲ 三角函数
  - ⑳ 尽管
  - ㉑ 分解成
  - ㉒ 布尔
  - ㉓ 等同于
  - ㉔ 大于
  - ㉕ 小于
  - ㉖ 超标量体系结构
  - ㉗ 多个

MIMD features often provide ALUs that can perform arithmetic on **vectors**<sup>⑧</sup> and **matrices**<sup>⑨</sup>.

### 3. Memory

Computer data **storage**<sup>⑩</sup>, often called storage or memory, is a technology consisting of computer components and recording media that are used to retain digital data. Magnetic-core memory shown as Figure 1-2 was the predominant form of random-access computer memory for 20 years between about 1955 and 1975.

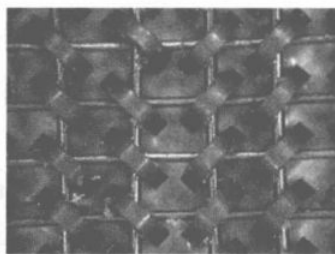


Figure 1-2 Magnetic core memory was the computer memory of choice

Magnetic core memory was popular main memory for computers through the 1960s, until it was replaced by semiconductor memory.

A computer's memory can be viewed as a list of cells into which numbers can be placed or read. Each cell has a numbered "address" and can store a single number. The computer can be instructed to "put the number 123 into<sup>⑪</sup> the cell numbered 1357" or to "add the number that is in cell 1357 to the number that is in cell 2468 and put the answer into cell 1595".<sup>注6</sup> The information stored in memory may represent practically anything. Letters, numbers, even computer instructions can be placed into memory with equal ease. Since the CPU does not differentiate between different types of information, it is the software's responsibility to give **significance**<sup>⑫</sup> to what the memory sees as **nothing but**<sup>⑬</sup> a series of numbers.

In almost all modern computers, each memory cell is set up to store binary numbers in groups of eight bits (called a byte). Each byte is able to represent 256 different numbers ( $2^8 = 256$ ); either from 0 to 255 or  $-128$  to  $+127$ . To store larger numbers, several **consecutive**<sup>⑭</sup> bytes may be used (typically, two, four or eight). When negative numbers

---

⑧ 向量

⑨ matrix 的复数形式, 矩阵

⑩ 存储

⑪ 将……放入

⑫ 意义

⑬ 只有, 只不过

⑭ 连续的

are required, they are usually stored in two's complement notation. Other arrangements are possible, but are usually not seen outside of specialized applications or historical **contexts**<sup>⑤</sup>. A computer can store any kind of information in memory if it can be represented numerically. Modern computers have billions or even trillions of bytes of memory.

The CPU contains a special set of memory cells called registers that can be read and written **too much more rapidly than**<sup>⑥</sup> the main memory area. There are typically between two and one hundred registers depending on the type of CPU. Registers are used for the most frequently needed data items to avoid having to access main memory every time data is needed.<sup>注7</sup> As data is constantly being worked on, reducing the need to access main memory (which is often slow compared to the ALU and control units) greatly increases the computer's speed.

Computer main memory comes in two **principal**<sup>⑦</sup> varieties: random-access memory or RAM and read-only memory or ROM. RAM can be read and written to anytime the CPU commands it, but ROM is pre-loaded with data and software that never changes, so the CPU can only read from it.<sup>注8</sup> ROM is typically used to store the computer's initial start-up instructions. In general, the contents of RAM are erased when the power to the computer is turned off, but ROM retains its data indefinitely. In a PC, the ROM contains a specialized program called the BIOS that orchestrates loading the computer's operating system from the hard disk drive into RAM whenever the computer is turned on or reset. In **embedded**<sup>⑧</sup> computers, which frequently do not have disk drives, all of the required software may be stored in ROM. Software stored in ROM is often called **firmware**<sup>⑨</sup>, because it is notionally more like hardware than software. Flash memory blurs the distinction between ROM and RAM, as it retains its data when turned off but is also rewritable. It is typically much slower than **conventional**<sup>⑩</sup> ROM and RAM however, so its use is **restricted**<sup>⑪</sup> to applications where high speed is unnecessary.

In more sophisticated computers there may be one or more RAM cache memories which are slower than registers but faster than main memory.<sup>注9</sup> Generally computers with this sort of **cache**<sup>⑫</sup> are designed to move frequently needed data into the cache automatically, often without the need for any **intervention**<sup>⑬</sup> on the programmer's part.

- 
- ⑤ 内容
  - ⑥ 快得多的……
  - ⑦ 重要的,主要的
  - ⑧ 嵌入式
  - ⑨ 固件
  - ⑩ 常规的
  - ⑪ 受约束的
  - ⑫ 高速缓存
  - ⑬ 干涉

#### 4. Input/Output (I/O)

I/O is the means by which a computer exchanges information with the outside world. Devices that provide input or output to the computer are called peripherals.<sup>①</sup> On a typical personal computer, peripherals include input devices like the keyboard and mouse, and output devices such as the display and printer.<sup>⑩</sup> **Hard disk drives**<sup>⑫</sup> shown as Figure 1-3, **floppy disk drives**<sup>⑬</sup> and **optical disc**<sup>⑭</sup> drives serve as both input and output devices. Computer networking is another form of I/O.



Figure 1-3 Hard disk drives are common storage devices used with computers

Often, I/O devices are complex computers in their own right with their own CPU and memory. A graphics processing unit might contain fifty or more tiny computers that perform the calculations necessary to display 3D graphics. Modern desktop computers contain many smaller computers that **assist**<sup>⑮</sup> the main CPU in performing I/O.

#### 5. Multitasking

While a computer may **be viewed as**<sup>⑯</sup> running one gigantic program stored in its main memory, in some systems it is necessary to give the appearance of running several programs **simultaneously**<sup>⑰</sup>. This is achieved by multitasking, i. e. having the computer switch rapidly between running each program **in turn**<sup>⑱</sup>.

One means by which this is done is with a special signal called an interrupt<sup>⑲</sup> which can periodically cause the computer to stop executing instructions where it was and do something else instead.<sup>⑳</sup> By remembering where it was executing prior to the interrupt, the computer can return to that task later. The interrupt generator might be causing several

- ① 外围设备
- ⑫ 硬盘驱动器
- ⑬ 软盘驱动器
- ⑭ 光盘
- ⑮ 帮助, 辅助
- ⑯ 看成
- ⑰ 同时
- ⑱ 依次
- ⑲ 打断, 中断

hundred interrupts per second, causing a program switch each time, if several programs are running “at the same time”. Since modern computers typically execute instructions **several orders of** ⑤ magnitude **faster than** ⑥ human perception, it may appear that many programs are running at the same time even though only one is ever executing in any given instant. This method of multitasking is sometimes termed “time-sharing” since each program is **allocated** ⑦ a “**slice**” ⑧ of time in turn.

Before the era of cheap computers, the principle use for multitasking was to allow many people to share the same computer.

Seemingly, multitasking would cause a computer that is switching between several programs to run more slowly—in **direct proportion to** ⑨ the number of programs it is running. However, most programs spend much of their time waiting for slow input/output devices to complete their tasks. If a program is waiting for the user to click on the mouse or press a key on the keyboard, then it will not take a “time slice” until the event it is waiting for has occurred. This frees **up time** ⑩ for other programs to execute so that many programs may be run at the same time without unacceptable speed **loss** ⑪.

## 6. Multiprocessing

Some computers are designed to **distribute** ⑫ their work across several CPUs in a **multiprocessing configuration** ⑬, a technique once **employed** ⑭ only in large and powerful machines such as supercomputers, **mainframe** ⑮ computers and servers. 注12 Multiprocessor and multi-core (multiple CPUs on a single integrated circuit) personal and laptop computers are now widely available, and are being increasingly used in lower-end markets as a result shown as Figure 1-4.

Supercomputers in particular often have highly unique architectures that differ significantly from the basic stored-program architecture and from general purpose computers. They often feature thousands of CPUs, customized high-speed interconnects, and specialized computing hardware. 注13 Such designs tend to be useful only for specialized tasks due to the large scale of program organization required to successfully utilize most of the available

- 
- ⑤ 几个数量级
  - ⑥ 更快的
  - ⑦ 分配
  - ⑧ 片
  - ⑨ 与……成正比
  - ⑩ 节省了时间
  - ⑪ 损失
  - ⑫ 分配
  - ⑬ 配置
  - ⑭ 使用
  - ⑮ 主机,大型机

resources at once. Supercomputers usually see usage in large-scale **simulation**<sup>⑩</sup>, graphics rendering, and cryptography applications, **as well as**<sup>⑪</sup> with other **so-called**<sup>⑫</sup> “**embarrassingly**<sup>⑬</sup> parallel” tasks.<sup>⑭</sup>



Figure 1-4 Cray designed many supercomputers that used multiprocessing heavily

## 7. Networking and the Internet

Computers have been used to **coordinate**<sup>⑮</sup> information between multiple locations since the 1950s. The U. S. military’s SAGE system was the first **large-scale**<sup>⑯</sup> example of such a system, which **led to a number of**<sup>⑰</sup> special-purpose **commercial**<sup>⑱</sup> systems like **Sabre**<sup>⑲</sup>.

In the 1970s, computer engineers at research institutions throughout the United States began to link their computers together using telecommunications technology.<sup>⑳</sup> This effort was funded by ARPA (now DARPA), and the computer network that it produced was called the ARPANET. The technologies that made the **ARPANet**<sup>㉑</sup> possible spread and evolved.

**In time**<sup>㉒</sup>, the network spread beyond academic and military institutions and became known as the Internet shown as Figure 1-5. The emergence of networking involved a **redefinition**<sup>㉓</sup> of the nature and boundaries of the computer. Computer operating systems and applications were modified to include the ability to define and access the resources of other computers on the network, such as peripheral devices, stored information, and the like,

- 
- ⑩ 模仿
  - ⑪ 也
  - ⑫ 称为
  - ⑬ 使人为难地
  - ⑭ 使协调
  - ⑮ 大规模地
  - ⑯ 导致许多的……
  - ⑰ 商业的
  - ⑱ 商业系统名称
  - ⑲ 互联网的始祖
  - ㉑ 最后
  - ㉒ 补救