

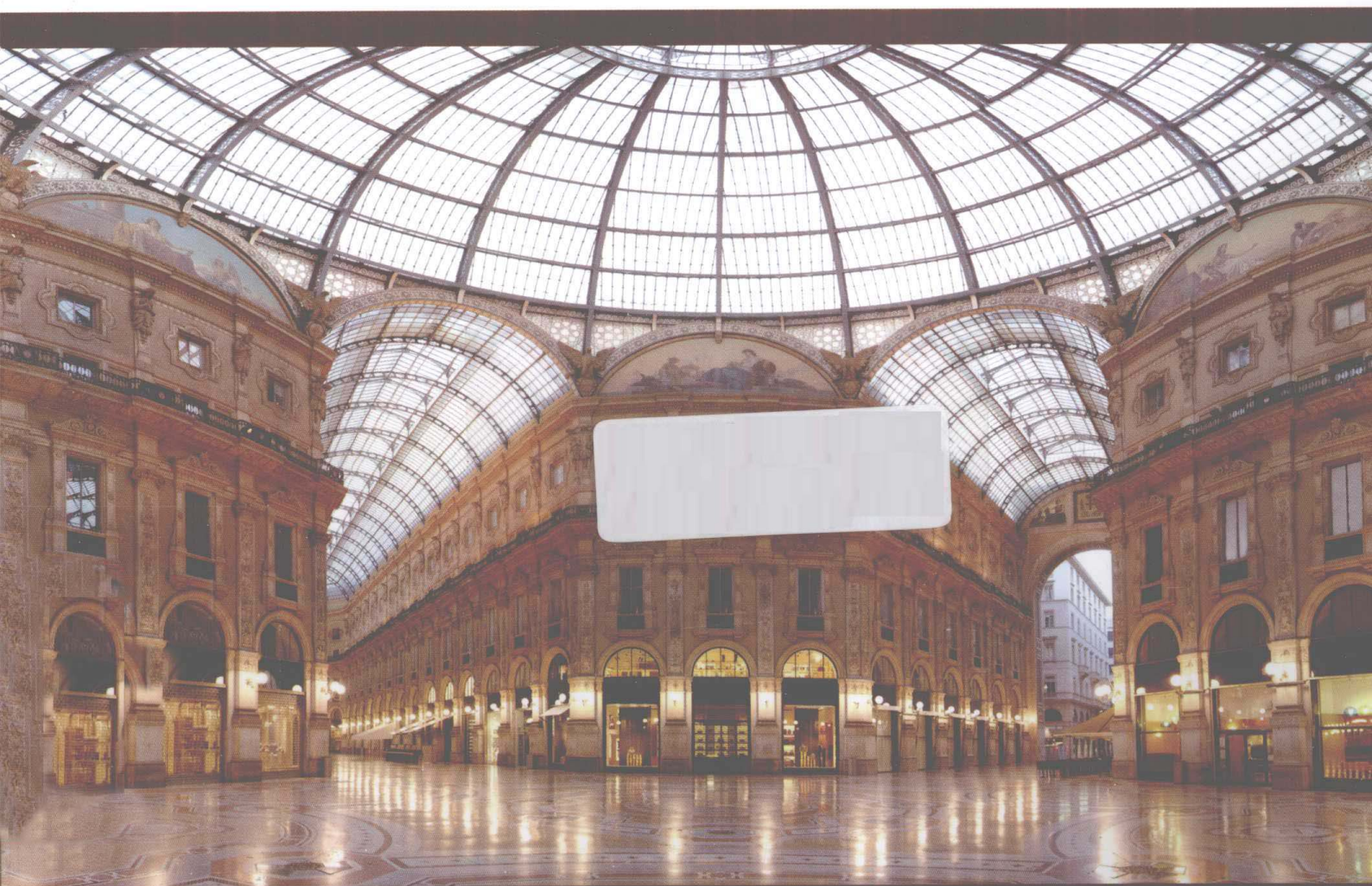
计算机英语

Computer English

(第4版)

(Fourth Edition)

刘 艺 王春生 等编



机械工业出版社
China Machine Press



Computer English

- 面向21世纪的计算机专业英语教材
- 按照最新大学英语教学大纲专业英语要求编写
- 精选最新英语时文和经典原版教材
- 悉心注解并附丰富练习
- 汇集计算机英语的大量专业词汇
- 版式新颖，图文并茂，便于自学
- 快速增进计算机原版文献资料阅读能力
- 有助于课堂教学、考试考证及出国发展
- 教辅材料完整配套，在线支持服务周到

• 相关教材 •



ISBN 978-7-111-39997-1
定价：35.00元



客服热线: (010) 88378991 88381000
购书热线: (010) 88326254 88379546 88381059
投稿热线: (010) 88379504

电子邮箱: caep@caep.com.cn
www.caep.com.cn
地址: 北京市西城区德胜门内大街2号

上架指导: 计算机/计算机英语

ISBN 978-7-111-42038-5



9 787111 420385 >

定价：45.00元

计算机英语

Computer English

(第4版)

(Fourth Edition)

刘 艺 王春生 等编



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

计算机英语 / 刘艺等编. —4 版. —北京: 机械工业出版社, 2013.4

ISBN 978-7-111-42038-5

I. 计… II. 刘… III. 电子计算机-英语 IV. TP3

中国版本图书馆 CIP 数据核字 (2013) 第 066975 号

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问 北京市展达律师事务所

本书涉及计算机与计算机科学的基础知识、计算机体系结构、计算机语言与编程、软件开发、软件过程、数据库、计算机网络、因特网、移动与云计算、计算机安全、计算机文化、电子商务等深刻影响我们生活的信息技术。本书以计算机领域英语时文和经典原版教材为基础, 通过大量精心挑选的阅读材料, 配以相应的注释和练习, 使读者能够快速掌握计算机领域的大量专业词汇以及相关的语法等, 并提高阅读和检索计算机原版文献资料的能力。

本书选材广泛、图文并茂, 采用双色印刷, 极大方便了读者的学习和查阅。书后还附有词汇表和缩略语表。本书可作为高等院校计算机及相关专业“计算机英语”课程的教材, 也可供参加计算机水平考试的考生、IT 行业的工程技术人员以及其他有需要的读者学习参考。

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑: 余洁

中国电影出版社印刷厂印刷

2013 年 5 月第 4 版第 1 次印刷

185mm×260mm·20 印张

标准书号: ISBN 978-7-111-42038-5

定 价: 45.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzsj@hzbook.com

前 言

英语作为计算机及 IT 业的行业性语言,有着其他语言所不能替代的功能。无论是学习最新的计算机技术,还是使用最新的计算机软硬件产品,都离不开对计算机英语的熟练掌握。正是为了适应这种要求,不少有实力、有远见的院校纷纷开设了计算机英语课。有些院校不仅把它作为计算机专业的必修课,还将其作为一门实用的选修课推广到其他专业。

本书是按照最新《大学英语教学大纲》对专业英语的要求,为开设计算机英语课程而编写的面向 21 世纪的教材。在满足计算机专业英语教学的同时,我们并没有过分沉溺于晦涩抽象的理论和专业术语之中,而是注重实际应用与调动学习兴趣。本书选材广泛,内容丰富,涉及计算机基础知识、硬件结构、程序设计、软件工程、应用开发、网络通信、信息安全、电子商务以及其他深刻影响着我们生活的信息技术。

值得一提的是,我们编写出版的《计算机英语》深受读者喜爱,已被全国上百所高校采用。经过 12 年来的多次修订改版,已经在全国热销了 20 万册,作为计算机专业英语教材其销量在同类图书中一直领先。为不断保持教学创新,跟踪计算机英语的发展,本教材与时俱进,不断升级完善。第 4 版更新了约 1/4 的文章和练习,并加强了教辅和售后服务。这次经过全面修订的《计算机英语(第 4 版)》更加适应深化计算机英语教学改革的需要。

这次修订主要完成以下工作:

- 课文更新 第 4 版替换了内容过时或不太合适的课文,约占全书总篇幅的 1/4。新增的课文紧跟技术潮流,把握主流趋势。既有涉及云计算这样的最新时文,也有关于计算机科学的经典论述。借此也引入了大量新的技术术语和缩略语,为读者尽快独立阅读英文资料,融入英文技术社区提供帮助。
- 对保留的课文进行修改 在本次修订中,我们对一些保留课文中的疏漏进行了改正,调整了部分注解,使之更加贴切。
- 保持全书的体系结构 本书参考 ACM 推荐的 CS0、CS1、CS2 课程体系,并保持第 3 版的 12 个教学单元不变。新版中去除了过时的文章,并调整和优化了内容结构,新增了移动与云计算单元。

本书在出版过程中得到机械工业出版社华章分社的鼎力支持,同时谈亮、刘春花、洪蕾、王珊珊、蒋丹丹、蔡敏、徐大琴、吴英等也为本书的编写和顺利出版付出了心血,在此一并表示感谢!

尽管本书在编写过程中,在资料的查核、术语的汉译、生词的注音以及文字的规范等方面都做了大量工作,但由于计算机领域的发展日新月异,许多新术语尚无确定的规范译法,加上编者水平有限,书中难免有不尽如人意之处,恳请广大读者不吝赐教。

编 者

2013 年 2 月于南京

E-mail: book4u@sina.com

计算机英语 BBS: <http://www.xici.net/b1119133/>

使用说明

1. 本教材共 12 单元，每单元包括三篇文章。课文 A 为精读材料，课文 B 和课文 C 为泛读材料。三篇课文均围绕同一主题。课文 A 一般为该主题的概述，课文 B 和课文 C 多就该主题的某一具体方面或具体例子展开讨论。课文长度一般为 1200~1800 词。对于课文 A，要求能正确理解和熟练掌握其内容。对于课文 B 和课文 C，要求能掌握中心大意，抓住主要事实。

2. 课文 A 配有四项练习，即“课文理解填空”、“词组中英文互译”、“完形填空”和“段落翻译”；课文 B 和课文 C 配有两项练习，即“课文理解填空”和“词组中英文互译”。各项练习均与课文内容和计算机专业紧密结合，旨在巩固和拓展学生所学内容。

3. 教育部 2007 年印发的《大学英语课程教学要求》将高等学校非英语专业本科生的英语教学要求分为三个层次，即“一般要求”、“较高要求”和“更高要求”，并规定：“一般要求是高等学校非英语专业本科毕业生应达到的基本要求。”专业英语课程按要求应放在大学英语（公共英语）教学完成之后。这就意味着大学本科生在学习专业英语之前应达到“一般要求”，但不一定达到“较高要求”或“更高要求”。因此，本教材的生词选注参照“一般要求”应掌握的词汇，所注生词计有两类：一类是通用词，即超出“一般要求”的词汇；另一类为计算机及相关专业词汇，是超出“一般要求”或“一般要求”未注明有关词义的词汇。

4. 生词均用蓝色粗体在课文中标出，并在课文旁边的文本框中进行注释，以便于阅读和记忆。相同生词原则上只在首次出现之处进行注释，但书后附有词汇表备查。每个生词一般标注一个发音，但有的常用异读音也标了出来，中间用逗号分隔；如系英美发音差异，英国发音在前，美国发音在后，中间用分号分隔。生词注音中的斜体音标表示该音可读可不读；短划（-）用于截同示异，代表与前面所注发音相同的部分。

5. 计算机英语的特点之一是大量使用缩略语。本教材对缩略语采取两种处理方式：一是在首次出现之处进行脚注；二是对大家熟知或其意思在文中不言自明的缩略语未加脚注。但是，无论哪种情况，所有缩略语均收入书后的缩略语表，以备查阅和方便记忆。另外，在计算机英语中，缩略语所代表的词组或术语在大小写上有比较随意和不一致的现象，本教材原则上尊重原文所采用的形式。

6. 关于计算机专业英语课的开设，建议本科生安排在第五学期至第七学期，研究生安排在第一学年。完成本教材的学习后应达到的要求，可参照教育部颁布的《大学英语教学大纲》（修订本）有关专业英语的规定。

编者

2013 年 2 月

目 录

前言

使用说明

Unit 1 Computer and Computer Science (计算机与计算机科学)	1
<i>Section A Computer Overview</i>	1
<i>Section B What Is Computer Science</i>	11
<i>Section C Categories of Computers</i>	18
Unit 2 Computer Architecture (计算机体系结构)	25
<i>Section A Computer Hardware</i>	25
<i>Section B Components of an Operating System</i>	32
<i>Section C System Organization</i>	39
Unit 3 Computer Language and Programming (计算机语言与编程)	47
<i>Section A Programming Language</i>	47
<i>Section B The Java Language</i>	56
<i>Section C Arrays</i>	63
Unit 4 Software Development (软件开发)	70
<i>Section A Computer Program</i>	70
<i>Section B Model Driven Development</i>	80
<i>Section C Distributed Systems</i>	87
Unit 5 Software Process (软件过程)	95
<i>Section A Software Process Models</i>	95
<i>Section B Software Testing Techniques</i>	105
<i>Section C What Is a Design Pattern</i>	111
Unit 6 Database (数据库)	117
<i>Section A Database Overview</i>	117
<i>Section B Maintaining Database Integrity</i>	127

Section C <i>Data Mining</i>	133
Unit 7 Computer Network (计算机网络)	141
Section A <i>Network Fundamentals</i>	141
Section B <i>A Guide to Network Topology</i>	150
Section C <i>Network Connecting Devices</i>	156
Unit 8 The Internet (因特网)	162
Section A <i>The Internet</i>	162
Section B <i>The Layered Approach to Internet Software</i>	171
Section C <i>Web Basics</i>	178
Unit 9 Mobile and Cloud Computing (移动与云计算)	186
Section A <i>Cloud Computing</i>	186
Section B <i>The Internet of Things</i>	196
Section C <i>Mobile Users</i>	204
Unit 10 Computer Security (计算机安全)	210
Section A <i>Computer Security</i>	210
Section B <i>Antivirus Software</i>	219
Section C <i>Types of Malicious Software</i>	225
Unit 11 Cyberculture (计算机文化)	231
Section A <i>Using E-Mail</i>	231
Section B <i>Designing Websites</i>	241
Section C <i>Social Issues of Computer Networks</i>	250
Unit 12 Electronic Commerce (电子商务)	255
Section A <i>Electronic Commerce</i>	255
Section B <i>Basic Functions of Electronic Commerce Software</i>	265
Section C <i>Online Shopping</i>	272
Glossary (词汇表)	279
Abbreviations (缩略语表)	311

Unit 1 Computer and Computer Science

(计算机与计算机科学)

Section A

Computer Overview



overview

/ 'əʊvəvju: /

n. 概述; 概观

calculation

/ ,kælkju'leɪʃən /

n. 计算

numeric(al)

/ nju:'merik(əl) /

a. 数字的; 数值的

banking / 'bæŋkɪŋ /

n. 银行业 (务)

I. Introduction

A computer is an electronic device that can receive a set of instructions, or program, and then carry out this program by performing **calculations** on **numerical** data or by manipulating other forms of information.

The modern world of high technology could not have come about¹ **except** for the development of the computer. Different types and sizes of computers find uses throughout society in the storage and handling of data, from secret governmental files to **banking** transactions to private

¹ *come about*: 发生, 产生。

transaction

/træn'zækʃən, -'sæk-/
n. 交易; 事务 (处理), 事项 (处理)

automation

/ɔ:tə'meɪʃən/
n. 自动化

conjecture

/kən'dʒektʃə/
n. 推测, 猜想

privacy

/'prɪvəsi, 'prai-, 'prai-/
n. 隐私; 秘密

precursor

/'pri:kə:sə, pri-/
n. 先驱

digit

/'dɪdʒɪt/
n. 数字

loom

/lu:m/ n. 织机

perforated

/'pɜ:fəreɪtɪd/
a. 穿孔的

statistician

/.stætɪ'stɪʃən/
n. 统计学家; 统计员

conceive

/kən'si:v/
v. (构) 想出

punched card

穿孔卡片

electrical contact

电触点

compile

/kəm'paɪl/
v. 汇编; 编译

census

/'sensəs/
n. 人口普查

analytic(al)

/.ænə'lɪtɪk(əl)/
a. 分析的

Analytical Engine

分析机, 解析机

household accounts. Computers have opened up a new era in manufacturing through the techniques of **automation**, and they have enhanced modern communication systems. They are essential tools in almost every field of research and applied technology, from constructing models of the universe to producing tomorrow's weather reports, and their use has in itself opened up new areas of **conjecture**. Database services and computer networks make available a great variety of information sources.¹ The same advanced techniques also make possible invasions of personal and business **privacy**. Computer crime has become one of the many risks that are part of the price of modern technology.

II. History

The first adding machine, a **precursor** of the digital computer, was devised in 1642 by the French scientist, mathematician, and philosopher Blaise Pascal². This device employed a series of ten-toothed wheels, each tooth representing a **digit** from 0 to 9. The wheels were connected so that numbers could be added to each other by advancing the wheels by a correct number of teeth. In the 1670s the German philosopher and mathematician Gottfried Wilhelm Leibniz³ improved on this machine by devising one that could also multiply.

The French inventor Joseph-Marie Jacquard⁴, in designing an automatic **loom**, used thin, **perforated** wooden boards to control the weaving of complicated designs. During the 1880s the American **statistician** Herman Hollerith⁵ **conceived** the idea of using perforated cards, similar to Jacquard's boards, for processing data. Employing a system that passed **punched cards** over **electrical contacts**, he was able to **compile** statistical information for the 1890 United States **census**.

1. The Analytical Engine

Also in the 19th century, the British mathematician and inventor

¹ *Database services and computer networks make available a great variety of information sources.*: 数据库服务和计算机网络使各种各样的信息源可供使用。这句话由于宾语 (a great variety of information sources) 较长, 因此将宾语的补语 (available) 放到了宾语的前面。文中下一句话也属同样情况。

² *Blaise Pascal*: 布莱斯·帕斯卡 (1623—1662), 法国数学家、物理学家、哲学家, 概率论创立者之一。

³ *Gottfried Wilhelm Leibniz*: 戈特弗里德·威廉·莱布尼兹 (1646—1716), 德国自然科学家、哲学家, 微积分、数理逻辑的先驱, 提出了二进制。

⁴ *Joseph-Marie Jacquard*: 约瑟夫-玛丽·雅卡尔 (1752—1834), 法国著名的织机工匠, 纹板提花机的主要改革家。

⁵ *Herman Hollerith*: 赫尔曼·何勒里斯 (1860—1929), 美国发明家和统计学家。

Difference Engine
差分机

input stream

输入(信息)流

deck /dek/

n. 卡片叠, 卡片组

printer /'prɪntə/

n. 打印机

analog(ue)

/ˈænələɡ/

a. 模拟的

shaft /ʃɑ:ft/

n. 轴

approximation

/əˌprɒksɪ'meɪʃən/

n. 近似(值)

torpedo /tɒˈpi:deɪ/

n. 鱼雷

submarine

/ˌsʌbməˈri:n/

n. 潜艇

bombsight

/ˈbɒmsaɪt/

n. 轰炸瞄准器

incorporate

/ɪnˈkɔ:pəreɪt/

v. 包含, 吸收; 把...
合并, 使并入

vacuum tube

真空管

crack /kræk/

v. 破译

encipher /ɪnˈsaɪfə/

v. 把...译成密码

Charles Babbage¹ worked out the principles of the modern digital computer. He conceived a number of machines, such as the **Difference Engine**, that were designed to handle complicated mathematical problems. Many historians consider Babbage and his associate, the mathematician Augusta Ada Byron², the true pioneers of the modern digital computer. One of Babbage's designs, the Analytical Engine, had many features of a modern computer. It had an **input stream** in the form of a **deck** of punched cards, a "store" for saving data, a "mill" for arithmetic operations, and a **printer** that made a permanent record.³ Babbage failed to put this idea into practice, though it may well have been technically possible at that date.

2. Early Computers

Analogue computers began to be built in the late 19th century. Early models calculated by means of rotating **shafts** and gears. Numerical **approximations** of equations too difficult to solve in any other way were evaluated with such machines. Lord Kelvin⁴ built a mechanical tide predictor that was a specialized analogue computer. During World Wars I and II, mechanical and, later, electrical analogue computing systems were used as **torpedo** course predictors in **submarines** and as **bombsight** controllers in aircraft. Another system was designed to predict spring floods in the Mississippi River⁵ basin.

3. Electronic Computers

During World War II, a team of scientists and mathematicians, working at Bletchley Park, north of London, created one of the first all-electronic digital computers: Colossus⁶. By December 1943, Colossus, which **incorporated** 1,500 **vacuum tubes**, was operational. It was used by the team headed by Alan Turing⁷, in the largely successful attempt to **crack** German radio messages **enciphered** in the Enigma code⁸.

¹ *Charles Babbage*: 查尔斯·巴比奇(1792—1871), 英国数学家和发明家。

² *Augusta Ada Byron*: 奥古斯塔·埃达·拜伦(1815—1852), 英国数学家, 诗人拜伦之女。

³ *It had an input stream in the form of a deck of punched cards, a "store" for saving data, a "mill" for arithmetic operations, and a printer that made a permanent record.*: 它有一个以一叠穿孔卡片的形式存在的输入流、一个保存数据的“仓库”、一个进行算术运算的“工厂”和一个产生永久性记录的打印机。

⁴ *Lord Kelvin*: 开尔文勋爵(1824—1907), 全名威廉·汤姆森·开尔文(William Thomson Kelvin), 英国物理学家, 发展了热力学理论, 创立了热力学绝对温标(即开尔文温标)。

⁵ *the Mississippi River*: 密西西比河, 发源于美国中北部的湖沼区, 南注墨西哥湾, 系美国主要河流。

⁶ *Colossus*: 该词读作 /kəˈlɒsəs/, 有“巨像”、“巨人”、“巨物”等意。

⁷ *Alan Turing*: 艾伦·图灵(1912—1954), 英国数学家和逻辑学家。

⁸ *Enigma code*: 恩尼格码, 德军在第二次世界大战期间采用的一种密码。

prototype

/'prəʊtətaɪp/

n. 原型; 样机

overshadow

/ˌəʊvə'shædəʊ/

v. 使相形见绌

integrator

/'ɪntɪɡreɪtə/

n. 积分器

patent /'peɪtənt/

n. 专利(权)

overturn

/ˌəʊvə'tɜːn/

v. 推翻; 废除

Independently of this, in the United States, a **prototype** electronic machine had been built as early as 1939, by John Atanasoff¹ and Clifford Berry² at Iowa State College³. This prototype and later research were completed quietly and later **overshadowed** by the development of the Electronic Numerical **Integrator** And Computer (ENIAC⁴) in 1945. ENIAC was granted a **patent**, which was **overturned** decades later, in 1973, when the machine was revealed to have incorporated principles first used in the Atanasoff-Berry Computer.

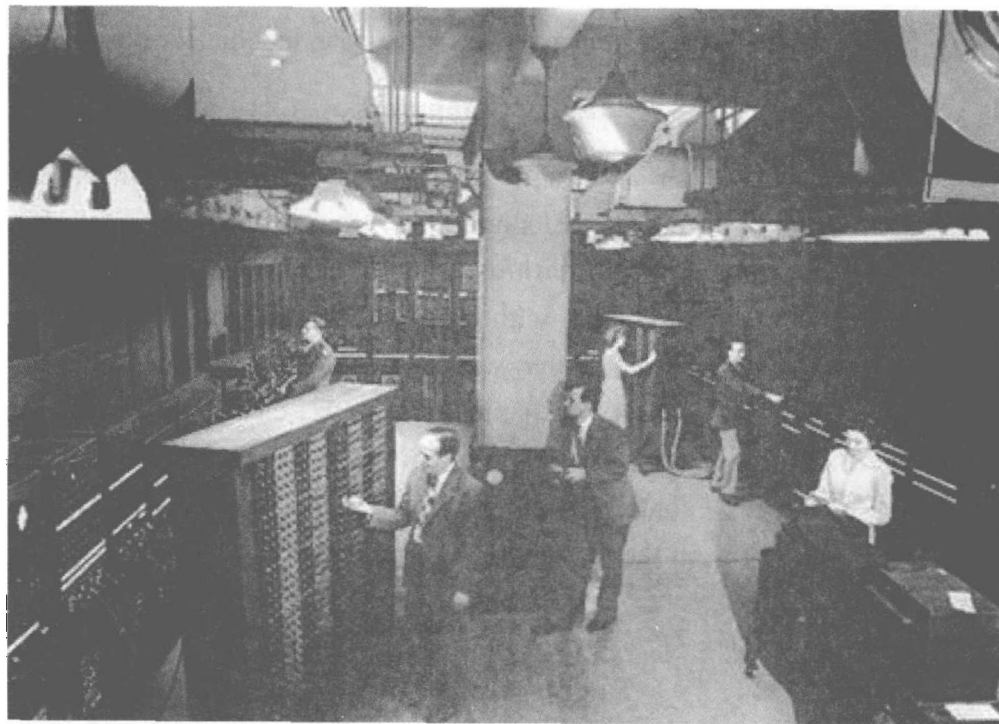


Figure 1A-1: ENIAC was one of the first fully electronic digital computers.

ENIAC (see Figure 1A-1) contained 18,000 vacuum tubes and had a speed of several hundred multiplications per minute, but originally its program was wired into the processor⁵ and had to be manually altered. Later machines were built with program storage, based on the ideas of the **Hungarian-American** mathematician John von Neumann⁶. The instructions,

Hungarian

/hʌŋ'ɡeəriən/

a. 匈牙利的

¹ *John Atanasoff*: 约翰·阿塔纳索夫(1903—1995), 美国物理学家。

² *Clifford Berry*: 克利福德·贝里(1918—1963), 美国物理学家。

³ *Iowa State College*: (美国)艾奥瓦州立学院。

⁴ *ENIAC*: 电子数字积分计算机, ENIAC 计算机(*Electronic Numerical Integrator And Computer* 的首字母缩略), 读作 /'i:niæk/。

⁵ *originally its program was wired into the processor*: 其程序最初是通过导线传送到处理器内的。

⁶ *John von Neumann*: 约翰·冯·诺依曼(1903—1957), 美籍匈牙利数学家, 对量子物理、数学逻辑和高速计算机的发展均有贡献。

memory /'meməri/

n. 存储器, 内存
paper-tape reader
纸带阅读器

execution

/ˌeksɪ'kjuʃən/

n. 执行, 运行

transistor

/træn'zistə, -'sis-/

n. 晶体管

advent /'ædvent/

n. 出现, 到来

versatile /'və:sətaɪl/

a. 多用途的,
通用的

logical element

逻辑元件

spacing /'speɪsɪŋ/

n. 间隔

integrated

/ˈɪntɪɡreɪtɪd/

a. 集成的, 综合的,
一体化的

integrated circuit

集成电路

fabricate

/ˈfæbrɪkeɪt/

v. 制作

substrate

/ˈsʌbstreɪt/

n. 衬底, 基底

microprocessor

/ˌmaɪkrəʊˈprəʊsesə/

n. 微处理器

microchip

/ˌmaɪkrəʊtʃɪp/

n. 微芯片

etch /etʃ/

v. 蚀刻

like the data, were stored within a “memory”, freeing the computer from the speed limitations of the **paper-tape reader** during **execution** and permitting problems to be solved without rewiring the computer.

The use of the **transistor** in computers in the late 1950s marked the **advent** of smaller, faster, and more **versatile logical elements** than were possible with vacuum-tube machines. Because transistors use much less power and have a much longer life, this development alone was responsible for the improved machines called second-generation computers. Components became smaller, as did inter-component **spacings**, and the system became much less expensive to build.

4. Integrated Circuits

Late in the 1960s the integrated circuit, or IC (see Figure 1A-2), was introduced, making it possible for many transistors to be **fabricated** on one **silicon substrate**, with interconnecting wires plated in place.¹ The IC resulted in a further reduction in price, size, and failure rate. The **microprocessor** became a reality in the mid-1970s with the introduction of the large-scale integrated (LSI) circuit and, later, the very large-scale integrated (VLSI) circuit (**microchip**), with many thousands of interconnected transistors **etched** into a single silicon substrate.

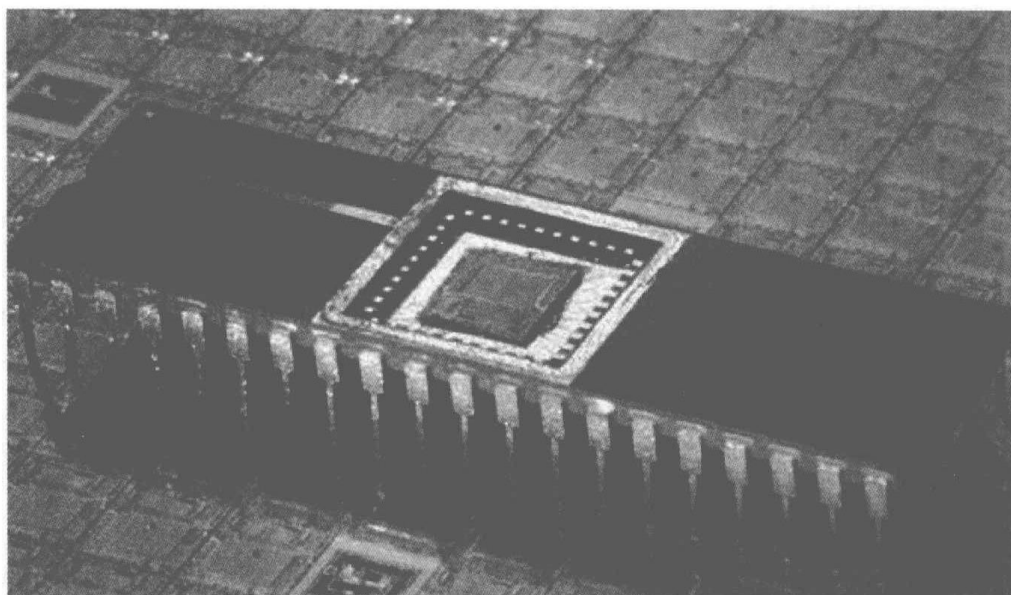


Figure 1A-2: An Integrated Circuit

To return, then, to the switching capabilities of a modern computer:

¹ *making it possible for many transistors to be fabricated on one silicon substrate, with interconnecting wires plated in place:* 从而有可能将许多晶体管制作在一块硅衬底上, 晶体管之间用覆镀在适当位置的导线相连接。

binary /'bainəri/
a. 二进制的
bit /bit/ n. 位, 比特
byte /bait/ n. 字节
datum /'deitəm/
n. ([复] **data**)
数据
graphics /'græfiks/
n. 图形, 图形显示
decimal /'desiməl/
a. 十进制的
switch /'switʃ/
n. 交换(设备);
开关; 转换
chip /tʃip/ n. 芯片
instruction set
指令集
ongoing /'ɒŋ.gəʊɪŋ/
a. 正在进行的,
不间断的
conceptually
/kən'septʃuəli/
ad. 概念上
microcomputer
/'maɪkrəʊkəm'pjʊ:tə/
n. 微型计算机
desktop /'desktp/
a. 桌面的; 台式(计
算机)的
laptop /'læptɒp/
n. 膝上型计算机
palmtop /'pɑ:mtɒp/
n. 掌上型计算机
workstation
/'wɜ:k.steɪʃən/
n. 工作站
minicomputer
/'minikəm'pjʊ:tə/
n. 小型计算机
mainframe
/'meɪnfreɪm/
n. 主机, 大型机
supercomputer
/'sju:pəkəm'pjʊ:tə/
n. 超级计算机,
巨型计算机
central processing
unit 中央处理器
bus /bʌs/ n. 总线

computers in the 1970s were generally able to handle eight switches at a time. That is, they could deal with eight **binary** digits, or **bits**, of data, at every cycle. A group of eight bits is called a **byte**, each byte containing 256 possible patterns of ONs and OFFs (or 1s and 0s). Each pattern is the equivalent of an instruction, a part of an instruction, or a particular type of **datum**, such as a number or a character or a **graphics** symbol. The pattern 11010010, for example, might be binary data—in this case, the **decimal** number 210—or it might be an instruction telling the computer to compare data stored in its **switches** to data stored in a certain memory-**chip** location.

The development of processors that can handle 16, 32, and 64 bits of data at a time has increased the speed of computers. The complete collection of recognizable patterns—the total list of operations—of which a computer is capable is called its **instruction set**. Both factors—the number of bits that can be handled at one time, and the size of instruction sets—continue to increase with the **ongoing** development of modern digital computers.

III. Hardware

Modern digital computers are all **conceptually** similar, regardless of size. Nevertheless, they can be divided into several categories on the basis of cost and performance: the personal computer or **microcomputer**, a relatively low-cost machine, usually of **desktop** size (though “**laptops**” are small enough to fit in a briefcase, and “**palmtops**” can fit into a pocket); the **workstation**, a microcomputer with enhanced graphics and communications capabilities that make it especially useful for office work; the **minicomputer**, generally too expensive for personal use, with capabilities suited to a business, school, or laboratory; and the **mainframe** computer, a large, expensive machine with the capability of serving the needs of major business enterprises, government departments, scientific research establishments, or the like (the largest and fastest of these are called **supercomputers**).

A digital computer is not a single machine: rather, it is a system composed of five distinct elements: (1) a **central processing unit**; (2) input devices; (3) memory storage devices; (4) output devices; and (5) a communications network, called a **bus**, which links all the elements of the system and connects the system to the external world.

IV. Programming

A program is a sequence of instructions that tells the hardware of a computer what operations to perform on data. Programs can be built into the hardware itself, or they may exist independently in a form known as software. In some specialized, or “**dedicated**”, computers the operating instructions are **embedded** in their **circuitry**; common examples are the microcomputers found in calculators, **wristwatches**, car engines, and **microwave ovens**. A general-purpose computer, on the other hand, although it contains some **built-in** programs (in ROM¹) or instructions (in the processor chip), depends on external programs to perform useful tasks. Once a computer has been programmed, it can do only as much or as little as the software controlling it at any given moment enables it to do. Software in widespread use includes a wide range of applications programs—instructions to the computer on how to perform various tasks.

V. Future Developments

One continuing trend in computer development is **microminiaturization**, the effort to compress more circuit elements into smaller and smaller chip space. Researchers are also trying to speed up circuitry functions through the use of **superconductivity**, the phenomenon of decreased electrical resistance observed in certain materials at very low temperatures.

The “fifth-generation” computer effort to develop computers that can solve complex problems in ways that might eventually merit the description “creative”² is another trend in computer development, the ideal goal being true **artificial intelligence**. One path actively being explored is parallel-processing computing, which uses many chips to perform several different tasks at the same time. One important parallel-processing approach is the **neural network**, which **mimics** the **architecture** of the nervous system. Another ongoing trend is the increase in computer **networking**, which now employs the worldwide data communications system of satellite and cable links to connect computers globally. There is also a great deal of research into the possibility of “optical” computers—hardware that processes not pulses of electricity but much faster pulses of light.

dedicated

/ˈdedikeɪtɪd/

a. 专用的

embed

/ɪmˈbed/

v. 把…嵌入

circuitry

/ˈsɜːkɪtri/

n. 电路

wristwatch

/ˈrɪstwɒtʃ/

n. 手表

microwave oven

微波炉

built-in

/ˈbɪltˈɪn/

a. 内置的, 内部的

microminiaturization

/ˈmaɪkrəʊˌmɪˌnɪˌtʃəraɪˌzeɪʃən/

n. 微小型化, 超小型化

superconductivity

/ˈsjuːpəˌkɒndʌkˈtɪvɪti/

n. 超导(电)性

artificial

intelligence

人工智能

neural

/ˈnjuərəl/

a. 神经的

neural network

神经网络

mimic

/ˈmɪmɪk/

v. 模仿

architecture

/ˈɑːkɪtektʃə/

n. 体系结构

networking

/ˈnet.wɜːkɪŋ/

n. 连网, 联网, 建网

¹ ROM: 只读存储器 (read-only memory 的首字母缩略), 读作 /rɒm/。

² merit the description “creative”: 用“创造性的”这个词来形容名副其实。

Exercises

I. Fill in the blanks with the information given in the text:

1. According to many historians, the true pioneers of the modern digital computer are _____ and _____.
2. A digital computer is generally made up of five distinct elements: a central processing unit, _____ devices, memory storage devices, _____ devices, and a bus.
3. The microprocessor is a central processing unit on a single chip. It was made possible in the mid-1970s with the introduction of the LSI circuit and the _____ circuit.
4. According to the text, modern digital computers can be divided into four major categories on the basis of cost and performance. They are microcomputers, _____, minicomputers, and _____.
5. The first electronic computers, such as Colossus and ENIAC created in Britain and the United States respectively, used _____ tubes, which later gave place to _____.
6. A program is a sequence of _____ that can be executed by a computer. It can either be built into the hardware or exist independently in the form of _____.
7. The smallest unit of information handled by a computer is bit, which is the abbreviation of binary _____. A group of _____ bits is called a(n) _____.
8. One of the trends in computer development is _____, the effort to compress more circuit elements into smaller and smaller _____ space.

II. Translate the following terms or phrases from English into Chinese and vice versa:

- | | |
|----------------------------|--------------|
| 1. artificial intelligence | 11. 数据的存储与处理 |
| 2. paper-tape reader | 12. 超大规模集成电路 |
| 3. optical computer | 13. 中央处理器 |

- | | |
|------------------------------|-----------|
| 4. neural network | 14. 个人计算机 |
| 5. instruction set | 15. 模拟计算机 |
| 6. parallel processing | 16. 数字计算机 |
| 7. difference engine | 17. 通用计算机 |
| 8. versatile logical element | 18. 处理器芯片 |
| 9. silicon substrate | 19. 操作指令 |
| 10. vacuum tube | 20. 输入设备 |

III. Fill in each of the blanks with one of the words given in the following list, making changes if necessary:

<i>microcomputer</i>	<i>computing</i>	<i>digital</i>	<i>base</i>
<i>advent</i>	<i>mode</i>	<i>circuit</i>	<i>significance</i>
<i>chip</i>	<i>appear</i>	<i>speed</i>	<i>transistor</i>
<i>minicomputer</i>	<i>combine</i>	<i>categorization</i>	<i>integration</i>

We can define a computer as a device that accepts input, processes data, stores data, and produces output. According to the _____ of processing, computers are either analog or _____. They can also be classified as mainframes, _____, workstations, or microcomputers. All else (for example, the age of the machine) being equal, this _____ provides some indication of the computer's _____, size, cost, and abilities.

Ever since the _____ of computers, there have been constant changes. First-generation computers of historic _____, such as *UNIVAC* (通用自动计算机), introduced in the early 1950s, were _____ on vacuum tubes. Second-generation computers, _____ in the early 1960s, were those in which _____ replaced vacuum tubes. In third-generation computers, dating from the 1960s, integrated _____ replaced transistors. In fourth-generation computers such as _____, which first appeared in the mid-1970s, large-scale _____ enabled thousands of circuits to be incorporated on one _____. Fifth-generation computers are expected to _____ very-large-scale integration with sophisticated approaches to _____, including artificial intelligence and true distributed processing.

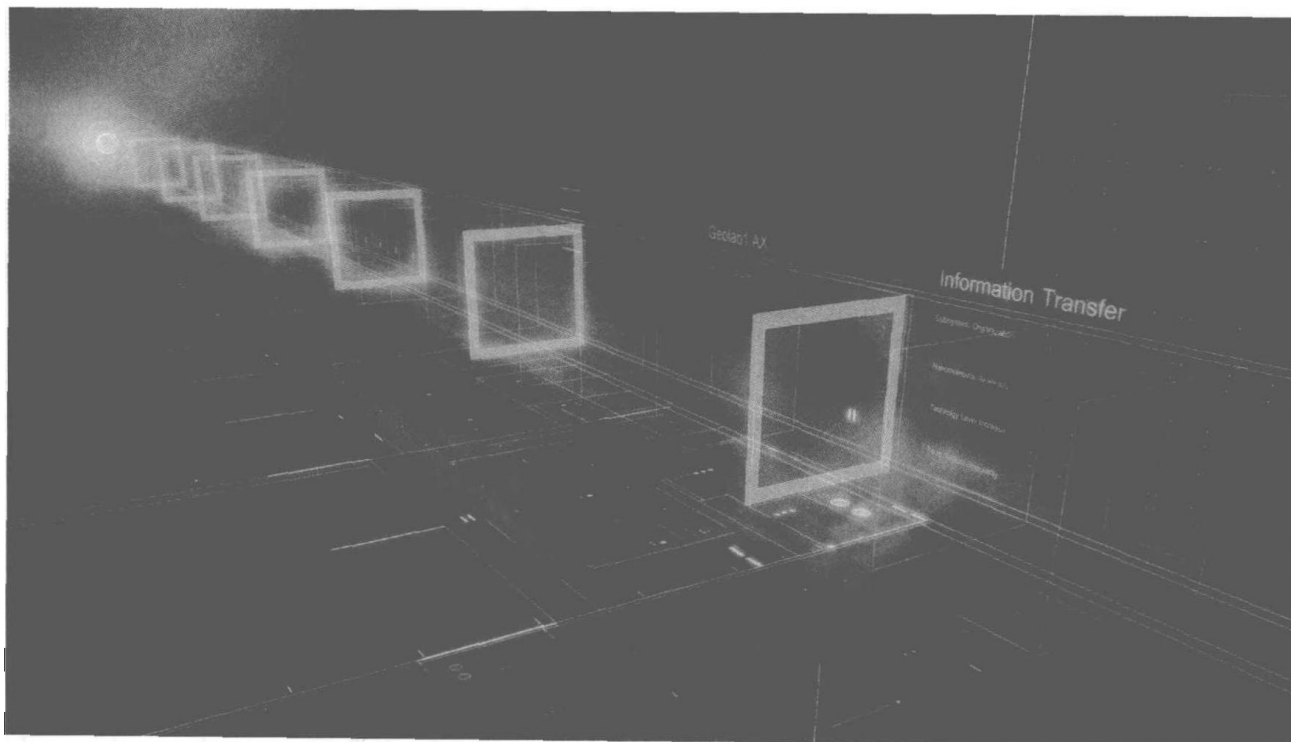
IV. Translate the following passage from English into Chinese:

Computers will become more advanced and they will also become easier to use. Improved speech recognition will make the operation of a

computer easier. *Virtual reality* (虚拟现实), the technology of interacting with a computer using all of the human senses, will also contribute to better human and computer *interfaces* (界面, 接口). Other, *exotic* (奇异的) models of computation are being developed, including biological computing that uses living organisms, *molecular* (分子的) computing that uses molecules with particular properties, and computing that uses deoxyribonucleic acid (DNA), the basic unit of *heredity* (遗传), to store data and carry out operations. These are examples of possible future computational platforms that, so far, are limited in abilities or are strictly theoretical. Scientists investigate them because of the physical limitations of miniaturizing circuits embedded in silicon. There are also limitations related to heat generated by even the tiniest of transistors.

Section B

What Is Computer Science



I. Introduction

Computer science is the study of the theory, experimentation, and engineering that form the basis for the design and use of computers—devices that automatically process information. Computer science traces its roots to work done by English mathematician Charles Babbage, who first proposed a programmable mechanical calculator in 1837. Until the advent of electronic digital computers in the 1940s, computer science was not generally distinguished as being separate from mathematics and engineering. Since then it has **sprouted** numerous branches of research that are unique to the discipline.

sprout / spraut /
v. 发芽; 长出

II. The Development of Computer Science

Early work in the field of computer science during the late 1940s and early 1950s focused on **automating** the process of making calculations for use in science and engineering. Scientists and engineers developed theoretical models of computation that enabled them to analyze how

automate
/ 'ɔ:təmeɪt /
v. 使自动化

overlap /ˌəʊvə'læp/
v. 重叠, 层叠
numerical analysis
数值分析

interface /'intəfeɪs/
n. 界面; 接口
application
/ˌæplɪ'keɪʃən/
n. 应用程序,
应用软件
mass-produce
/'mæsprə.dju:s/
v. (使用机器) 大量
生产

existing /ɪg'zɪstɪŋ/
a. 现存的, 现有的,
现行的
integral /'ɪntɪgrəl/
a. 构成整体所必需
的, 基本的
strive /straɪv/
v. 努力; 力争
speculative
/'spekjʊlətɪv, -ləɪ/
a. 推测的; 纯理论
的; 投机性的
viable /'vaɪəbl/
a. 切实可行的; 可
实施的
underlie /ˌʌndə'laɪ/
v. 构成...的基础
(或起因); 支持

efficient different approaches were in performing various calculations. Computer science **overlapped** considerably during this time with the branch of mathematics known as **numerical analysis**, which examines the accuracy and precision of calculations.

As the use of computers expanded between the 1950s and the 1970s, the focus of computer science broadened to include simplifying the use of computers through programming languages—artificial languages used to program computers, and operating systems—computer programs that provide a useful **interface** between a computer and a user. During this time, computer scientists were also experimenting with new **applications** and computer designs, creating the first computer networks, and exploring relationships between computation and thought.

In the 1970s, computer chip manufacturers began to **mass-produce** microprocessors—the electronic circuitry that serves as the main information processing center in a computer. This new technology revolutionized the computer industry by dramatically reducing the cost of building computers and greatly increasing their processing speed. The microprocessor made possible the advent of the personal computer, which resulted in an explosion in the use of computer applications. Between the early 1970s and 1980s, computer science rapidly expanded in an effort to develop new applications for personal computers and to drive the technological advances in the computing industry. Much of the earlier research that had been done began to reach the public through personal computers, which derived most of their early software from **existing** concepts and systems.

Computer scientists continue to expand the frontiers of computer and information systems by pioneering the designs of more complex, reliable, and powerful computers; enabling networks of computers to efficiently exchange vast amounts of information; and seeking ways to make computers behave intelligently. As computers become an increasingly **integral** part of modern society, computer scientists **strive** to solve new problems and invent better methods of solving current problems.

The goals of computer science range from finding ways to better educate people in the use of existing computers to highly **speculative** research into technologies and approaches that may not be **viable** for decades. **Underlying** all of these specific goals is the desire to better the human condition today and in the future through the improved use of

information.¹

III. Theory and Experiment

Computer science is a combination of theory, engineering, and experimentation. In some cases, a computer scientist develops a theory, then engineers a combination of computer hardware and software based on that theory, and experimentally tests it. An example of such a theory-driven approach is the development of new software engineering tools that are then evaluated in actual use. In other cases, experimentation may result in new theory, such as the discovery that an artificial neural network exhibits behavior similar to neurons in the brain, leading to a new theory in neurophysiology.

neuron /'njuəron/

n. 神经元,

神经细胞

neurophysiology

/'njuərou.fizi'ɒlədʒi/

n. 神经生理学

unforeseen

/ˌʌnfɔ:'si:n/

a. 未预见到的;

意料之外的

It might seem that the predictable nature of computers makes experimentation unnecessary because the outcome of experiments should be known in advance. But when computer systems and their interactions with the natural world become sufficiently complex, unforeseen behavior can result. Experimentation and the traditional scientific method are thus key parts of computer science.

IV. Major Branches of Computer Science

Computer science can be divided into four main fields: software development, computer architecture (hardware), human-computer interfacing (the design of the most efficient ways for humans to use computers), and artificial intelligence (the attempt to make computers behave intelligently). Software development is concerned with creating computer programs that perform efficiently. Computer architecture is concerned with developing optimal hardware for specific computational needs. The areas of artificial intelligence (AI²) and human-computer interfacing often involve the development of both software and hardware to solve specific problems.

interfacing

/ˌɪntəˌfeɪsɪŋ/

n. 接口技术

optimal /'ɒptɪmə/

a. 最适的, 最优的, 最佳的

computational

/ˌkɒmpju:'teɪʃənəl/

a. 计算(机)的

1. Software Development

In developing computer software, computer scientists and engineers study various areas and techniques of software design, such as the best

¹ Underlying all of these specific goals is the desire to better the human condition today and in the future through the improved use of information.: 这是一个倒装句, 正常语序应为 The desire to better ... information is underlying all of these specific goals.. 主语部分的中心词 desire 后面跟了一个较长的不定式短语作定语, 采用倒装语序是为了避免头重脚轻。

² AI: 人工智能 (artificial intelligence 的首字母缩略)。

algorithm

/'ælgəriðəm/

n. 算法

retrieve /ri'tri:v/

v. 检索

types of programming languages and **algorithms** to use in specific programs, how to efficiently store and **retrieve** information, and the computational limits of certain software-computer combinations. Software designers must consider many factors when developing a program. Often, program performance in one area must be sacrificed for the sake of the general performance of the software. For instance, since computers have only a limited amount of memory, software designers must limit the number of features they include in a program so that it will not require more memory than the system it is designed for can supply.

robust /rəu'bʌst ,

'rəubʌst/

a. 健壮的, 坚固的

life cycle

生命周期, 生存周期, 寿命周期

specification

/ˌspesifi'keɪʃən/

n. 明确说明; [常作~s] 规格, 规范

implementation

/ˌimplimen'teɪʃən/

n. 实现, 执行

Software engineering is an area of software development in which computer scientists and engineers study methods and tools that facilitate the efficient development of correct, reliable, and **robust** computer programs. Research in this branch of computer science considers all the phases of the software **life cycle**, which begins with a formal problem **specification**, and progresses to the design of a solution, its **implementation** as a program, testing of the program, and program maintenance. Software engineers develop software tools and collections of tools called programming environments to improve the development process. For example, tools can help to manage the many components of a large program that is being written by a team of programmers.

2. Computer Architecture

Computer architecture is the design and analysis of new computer systems. Computer architects study ways of improving computers by increasing their speed, storage capacity, and reliability, and by reducing their cost and power consumption. Computer architects develop both software and hardware models to analyze the performance of existing and proposed computer designs, and then use this analysis to guide the development of new computers. They are often involved with the engineering of a new computer because the accuracy of their models depends on the design of the computer's circuitry. Many computer architects are interested in developing computers that are specialized for particular applications such as image processing, signal processing, or the control of mechanical systems. The **optimization** of computer architecture to specific tasks often yields higher performance, lower cost, or both.¹

optimization

/ˌoptimaɪ'zeɪʃən/

n. 最佳化,
(最) 优化

¹ *The optimization of computer architecture to specific tasks often yields higher performance, lower cost, or both.*: 按照特定任务对计算机体系结构进行的优化, 常常带来性能的提高、成本的降低或两者。本句中的介词 to 表示“按”、“按照”。

3. Artificial Intelligence

Artificial intelligence (AI) research seeks to enable computers and machines to mimic human intelligence and **sensory** processing ability, and models human behavior with computers to improve our understanding of intelligence. The many branches of AI research include machine learning, inference, **cognition**, knowledge representation, problem solving, case-based reasoning, natural language understanding, speech recognition, **computer vision**, and artificial neural networks.

4. Robotics

Another area of computer science that has found wide practical use is robotics—the design and development of computer controlled mechanical devices. Robots range in **complexity** from toys to **automated** factory assembly lines, and relieve humans from tedious, **repetitive**, or dangerous tasks. Robots are also employed where requirements of speed, precision, consistency, or cleanliness exceed what humans can accomplish. **Roboticians**—scientists involved in the field of robotics—study the many aspects of controlling robots. These aspects include modeling the robot's physical properties, modeling its environment, planning its actions, directing its mechanisms efficiently, using **sensors** to provide feedback to the controlling program, and ensuring the safety of its behavior. They also study ways of simplifying the creation of control programs. One area of research seeks to provide robots with more of the **dexterity** and adaptability of humans, and is closely associated with AI.

5. Human-Computer Interfacing

Human-computer interfaces provide the means for people to use computers. An example of a human-computer interface is the keyboard, which lets humans enter commands into a computer and enter text into a specific application. The diversity of research into human-computer interfacing corresponds to the diversity of computer users and applications. However, a **unifying** theme is the development of better interfaces and experimental evaluation of their effectiveness. Examples include improving computer access for people with **disabilities**, simplifying program use, developing **three-dimensional** input and output devices for **virtual reality**, improving handwriting and speech recognition, and developing **head-up displays** for aircraft instruments in which critical information such as speed, altitude, and **heading** are displayed on a screen in front of the pilot's

sensory /'sensəri /

a. 感觉的; 感官的

cognition

/kəg'nɪʃən /

n. 认识, 认知

computer vision

计算机视觉

robotics /rəu'botiks /

n. 机器人学;

机器人技术

complexity

/kəm'pleksɪti /

n. 复杂性, 错综性

automated

/ˈɔ:təmeɪtɪd /

a. 自动化的

repetitive /ri'petɪtɪv /

a. 重复的

roboticist

/rəu'botɪsɪst /

n. 机器人 (制造操作) 专家

sensor /'sensə /

n. 传感器;

敏感元件

dexterity

/dek'sterɪti /

n. 灵巧, 敏捷

unify /'ju:nɪfaɪ /

v. 使联合, 统一;
使一致

disability

/ˌdɪsə'bɪlɪti /

n. 无能力; 残疾

three-dimensional

/ˈθri:di'menʃənəl /

a. 三维的, 立体的

virtual /'vɜ:tʃuəl /

a. 虚拟的

virtual reality

虚拟现实

head-up display

前导显示器, 平视
显示器

heading /'hedɪŋ /

n. 航向

visualization

/ˌvɪʒʊəlaɪ'zeɪʃən/

n. 可视化, 直观化

graphic(al)

/'græfɪk(əl)/

a. 图形的, 图示的, 图解的

comprehend

/ˌkɒmpri'hend/

v. 理解, 领会

linguistics

/lɪŋ'ɡwɪstɪks/

n. 语言学

physiologist

/ˌfɪzi'ɒlədʒɪst/

n. 生理学家

collaboration

/kə'læbə'reɪʃən/

n. 合作; 协作

interdisciplinary

/ˌɪntə'dɪsɪplɪnəri/

a. 学科间的, 跨学科的

window. One area of research, called **visualization**, is concerned with **graphically** presenting large amounts of data so that people can **comprehend** its key properties.

V. Connection of Computer Science to Other Disciplines

Theoretical computer science draws many of its approaches from mathematics and logic. Research in numerical computation overlaps with mathematics research in numerical analysis. Computer architects work closely with the electrical engineers who design the circuits of a computer.

Beyond these historical connections, there are strong ties between AI research and psychology, neurophysiology, and linguistics. Human-computer interface research also has connections with psychology. Roboticians work with both mechanical engineers and physiologists in designing new robots.

Computer science also has indirect relationships with virtually all disciplines that use computers. Applications developed in other fields often involve **collaboration** with computer scientists, who contribute their knowledge of algorithms, data structures, software engineering, and existing technology. In return, the computer scientists have the opportunity to observe novel applications of computers, from which they gain a deeper insight into their use. These relationships make computer science a highly **interdisciplinary** field of study.

Exercises

I. Fill in the blanks with the information given in the text:

1. Computer science, which traces its roots to work done by English mathematician Charles Babbage, is a combination of theory, engineering, and _____.

2. Computer science can be divided into four main fields: software development, computer architecture, human-computer _____, and artificial intelligence.

3. Computer science is said to be a highly _____ field of study because it has strong ties with many disciplines and indirect relationships with virtually all disciplines that use computers.

4. According to the text, it was the _____ that made possible the advent of the personal computer.

II. Translate the following terms or phrases from English into Chinese and vice versa:

- | | |
|------------------------------|------------|
| 1. artificial neural network | 6. 数值分析 |
| 2. computer architecture | 7. 程序设计环境 |
| 3. robust computer program | 8. 数据结构 |
| 4. human-computer interface | 9. 存储和检索信息 |
| 5. knowledge representation | 10. 虚拟现实 |

Section C

Categories of Computers



categorize

/ˈkætɪɡəraɪz/

v. 将...分类

criterion

/kraɪˈtɪəriən/

n. ([复] -ria / -ria / 或 -rions) 标准, 准则

differentiate

/ˌdɪfəˈrenʃieɪt/

v. 区分, 区别

Categorizing computers is a way of grouping them according to **criteria** such as usage, cost, size, and capability. At one time, it was possible to define three distinct categories of computers. Mainframes were housed in large, closet-sized metal frames. Minicomputers were smaller, less expensive, and less powerful computers that were able, nevertheless, to provide adequate computing power for small businesses. Microcomputers were clearly **differentiated** from computers in other categories because their CPUs¹ consisted of a single microprocessor chip.

Today, microprocessors are no longer a distinction between computer categories because just about every computer uses one or more microprocessors as its CPU. The term minicomputer has fallen into disuse

¹ CPU: 中央处理器 (central processing unit 的首字母缩略)。

server /'sə:və/

n. 服务器;
服务程序

abbreviate

/ə'brɪ:vieɪt/

v. 缩写

typically /'tɪpɪkəli/

ad. 一般地, 通常

form factor

(电子产品等的)
物理尺寸和形状;
规格

imaging /'ɪmɪdʒɪŋ/

n. 成像(技术)

computer-aided design

计算机辅助设计

animated

/ˈænɪmeɪtɪd/

a. 动画(片)的;
活生生的

videogame

/ˌvɪdɪəʊɡeɪm/

n. 电子游戏

console /'kɒnsəʊl/

n. 控制台, 操纵台

videogame console

电子游戏机

originate

/ə'ɪdʒɪneɪt/

v. 发源; 产生

joystick /'dʒɔɪstɪk/

n. 控制杆, 操纵杆,
游戏杆

add-on /'ædɒn/

n. 附加物;
附加装置

niche /nɪʃ, niːʃ/

n. 缝隙市场, “利基”市场

and the terms microcomputer and mainframe are used with less and less frequency. Experts don't necessarily agree on the categories or the devices placed in each category, but commonly used computer categories include personal computers, servers, mainframes, and supercomputers.

I. Personal Computers

The term personal computer is sometimes abbreviated as PC. However, PC can also refer to a specific type of personal computer that descended from the original IBM¹ PC and runs Windows software. A personal computer is a microprocessor-based computing device designed to meet the computing needs of an individual. It typically provides access to a wide variety of computing applications, such as word processing, photo editing, and e-mail. Personal computers are available as desktop or portable models, and in a variety of form factors, a term that refers to the dimensions of the unit that holds the computer circuitry².

The term workstation has two meanings. It can simply refer to an ordinary personal computer that is connected to a network. A second meaning refers to powerful desktop computers used for high-performance tasks, such as medical imaging and computer-aided design, that require a lot of processing speed. Some workstations contain more than one microprocessor, and most have circuitry specially designed for creating and displaying three-dimensional and animated graphics. Workstations typically cost a bit more than an average personal computer.

A videogame console is not generally referred to as a personal computer because of its history as a dedicated game device. Videogame consoles originated as simple digital devices that connected to a TV set and provided only a pair of joysticks for input. Today's videogame consoles contain microprocessors that are equivalent to any found in a fast personal computer, and they are equipped to produce graphics that rival those on sophisticated workstations. Add-ons such as keyboards, DVD³ players, and Internet access make it possible to use a videogame console to watch DVD movies, send and receive e-mail, and participate in online activities such as multiplayer games. Despite these features, videogame consoles fill a specialized niche and are not considered a replacement for a personal computer.

¹ IBM: (美国)国际商用机器公司(International Business Machines 的首字母缩略)。

² the dimensions of the unit that holds the computer circuitry: 容纳计算机电路的部件的尺寸规格。

³ DVD: 数字(激光)视盘(digital video disc 的首字母缩略)。

II. Servers

In the computer industry, the term server has several meanings. It can refer to computer hardware, to a specific type of software, or to a combination of hardware and software. In any case, the purpose of a server is to serve computers on a network (such as the Internet or a home network) by supplying them with data.

Any software or digital device, such as a computer, that requests data from a server is referred to as a **client**. For example, on a network, a server might respond to a client's request for a **Web page**. Another server might handle the steady stream of e-mail that travels among clients from all over the Internet. A server might also allow clients within a network to share files or access a centralized printer.

Remarkably, just about any personal computer, workstation, mainframe, or supercomputer can be **configured** to perform the work of a server. That fact should emphasize the concept that a server does not require a specific type of hardware. **Nonetheless**, computer manufacturers such as IBM and Dell¹ offer devices called **blade servers** and **storage servers** that are especially suited for storing and distributing data on a network. Server prices vary, depending on **configuration**, but tend to be more similar to workstation prices than personal computer prices. Despite impressive performance on server-related tasks, these machines do not offer features such as **sound cards**, DVD players, and other fun **accessories**, so they are not a suitable alternative to a personal computer.

III. Mainframes

A mainframe computer (or simply a mainframe) is a large and expensive computer capable of **simultaneously** processing data for hundreds or thousands of users. Mainframes are generally used by businesses or governments to provide centralized storage, processing, and management for large amounts of data. Mainframes remain the computer of choice² in situations where reliability, data security, and centralized control are necessary.

The price of a mainframe computer typically starts at \$100,000 and can easily exceed \$1 million. Its main processing circuitry is housed in a

client /'klaɪənt/

n. 客户机;
客户程序

Web page

(万维)网页

configure

/kən'fɪɡə/

v. 配置

nonetheless

/ˌnʌnðə'les/

ad. 尽管如此, 然而

blade server

刀片服务器

storage server

存储服务器

configuration

/kən.fɪɡjə'reɪʃən/

n. 配置

sound card

声卡

accessory

/ək'sesəri/

n. 附件, 配件

simultaneous

/ˌsɪmʌl'teɪniəs; ,sai-/

a. 同时(发生)的

¹ Dell: 戴尔公司 (Dell Inc.), 开发、制造和销售计算机及相关产品的著名公司, 成立于 1984 年, 总部设在美国得克萨斯州朗德罗克 (Round Rock)。

² of choice: 应先选择的, 首选的, 广受欢迎的。

closet-sized cabinet; but after large components are added for storage and output, a mainframe computer system can fill a **good-sized** room.

good-sized

/ˈɡʊdˈsaɪzd/

a. (相当) 大的;
宽大的

compute-intensive

/kəmˈpjʊːtɪnˈtensɪv/

a. 计算密集型的

molecular

/məˈlɛkjʊlə/

a. 分子的

data point

数据点

simulate /ˈsɪmjuleɪt/

v. 模拟, 仿真

simulation

/ˌsɪmjʊˈleɪʃən/

n. 模拟, 仿真

tornado /tɔːˈneɪdəʊ/

n. 龙卷风

smartphone

/ˈsmɑːtfəʊn/

n. 智能手机

programmability

/ˌprəʊgræməˈbɪlɪti/

n. 可编程(序)性

versatility

/ˌvɜːsəˈtɪlɪti/

n. 多用途; 通用性

customarily

/ˈkʌstəmərɪli; ˌkʌstə-

ˈmerɪli/

ad. 习惯上, 通常

qwerty /ˈkwɜːti/

n. 〈口〉标准打字机
键盘(亦作 qwerty
keyboard)

touch-sensitive

/ˈtʌtʃˌsensɪtɪv/

a. 触敏的

touch-sensitive

screen 触摸屏

synchronize

/ˈsɪŋkrənaɪz/

v. (使) 同步; (使)
协调

IV. Supercomputers

A computer falls into the supercomputer category if it is, at the time of construction, one of the fastest computers in the world. Because of their speed, supercomputers can tackle complex tasks and **compute-intensive** problems that just would not be practical for other computers. A compute-intensive problem is one that requires massive amounts of data to be processed using complex mathematical calculations. **Molecular** calculations, atmospheric models, and climate research are all examples of projects that require massive numbers of **data points** to be manipulated, processed, and analyzed.

Common uses for supercomputers include breaking codes, modeling worldwide weather systems, and **simulating** nuclear explosions. One impressive **simulation**, which was designed to run on a supercomputer, tracked the movement of thousands of dust particles as they were tossed about by a **tornado**.

At one time, supercomputer designers focused on building specialized, very fast, and very large CPUs. Today, most supercomputer CPUs are constructed from thousands of microprocessors. Of the 500 fastest supercomputers in the world, the majority use microprocessor technology.

V. PDAs¹, Smartphones, and Portable Media Players

Handheld digital devices incorporate many computer characteristics. They accept input, produce output, process data, and include storage capabilities. Handheld devices vary in their **programmability** and their **versatility**. Technically, most of these devices could be classified as computers, but they are **customarily** referred to by function, for example as PDAs, smartphones, and portable media players.

A PDA (personal digital assistant) is a pocket-sized digital appointment book with a small **qwerty** keyboard or a **touch-sensitive screen**, designed to run on batteries and be used while holding it. PDAs **synchronize** appointment data and contact lists with desktop computers by exchanging data over a dedicated wired or wireless connection. Originally, PDAs were not equipped for voice communications, which distinguished them from

¹ PDA: 个人数字助理 (personal digital assistant 的首字母缩略)。

cell phone

蜂窝电话, 移动电话,
手机

handheld computer

手持式计算机,
掌上电脑

full-featured

/ˈfʊlˈfi:tʃəd/

a. 全特征的, 全功
能的

scaled-down

/ˈskeildˈdaʊn/

a. 按比例缩减的

spreadsheet

/ˈspredʃi:t/

n. 电子表格, 电子
数据表

keypad /ˈki:pæd/

n. 小键盘, 袖珍键
盘; 键区

messaging

/ˈmesɪdʒɪŋ/

n. 消息接发, 通信

personal organizer

备忘录, 记事本

cell phones.

A PDA enhanced with features such as removable storage, e-mail, Web access, voice communications, built-in camera, and GPS¹ is sometimes called a **handheld computer**. These enhanced PDAs also offer a variety of application software, but they do not typically run the same **full-featured** software versions as personal computers. Instead, they run special **scaled-down** versions of word processing, **spreadsheet**, and other application software. Examples of handheld computers include the Apple iPhone², Palm Pre³, BlackBerry Curve⁴, and Hewlett-Packard iPAQ⁵.

Smartphones descended from basic cell phones that were originally designed exclusively for voice communications. In addition to voice communications, smartphones include features such as full qwerty **keypad**, **text messaging**, e-mail, Web access, removable storage, camera, FM⁶ radio, digital music player, and software options for games, financial management, **personal organizer**, GPS, and maps.

Smartphones contain a microprocessor and have many characteristics of computers. They are not, however, usually referred to as computers because of their origins as special-purpose devices with keypad input and limited programmability.

The iPod⁷ was designed as a portable music player and enhanced versions of the device now store and play video and photos as well. iPods and similar devices are classified as portable media players because their main strength is playing music, showing videos, and storing photos. Like other handheld digital devices, these players have many computer characteristics. An iPod, for example, contains a microprocessor, accepts input, has significant storage capacity on its built-in **hard disk**, and outputs stored music, video, and images. Most portable media players, however, have very limited programmability. They are not designed for users to add

hard disk

硬(磁)盘

¹ GPS: 全球(卫星)定位系统(Global Positioning System的首字母缩略)。

² Apple iPhone: 苹果 iPhone 手机。苹果公司(Apple Inc.), 原称苹果计算机公司(Apple Computer, Inc.), 1976 年创立, 总部位于美国加利福尼亚州丘珀蒂诺(Cupertino)。

³ Palm Pre: 奔迈 Pre 手机。奔迈公司(Palm, Inc.) 创立于 1992 年, 2010 年被惠普公司收购。

⁴ BlackBerry Curve: 黑莓 Curve 手机, 系 Research In Motion (RIM) 公司生产。RIM 公司是一家加拿大通信公司, 创立于 1999 年, 总部设在加拿大安大略省滑铁卢市。RIM 公司致力于开发可支持多种无线网络的集成软件、硬件和服务, 主要产品为 BlackBerry® 智能手机。

⁵ Hewlett-Packard iPAQ: 惠普 iPAQ 手机。惠普公司创立于 1939 年, 总部设在美国加利福尼亚州帕洛阿尔托(Palo Alto)。

⁶ FM: 调频(广播系统), 频率调制(frequency modulation的首字母缩略)。

⁷ iPod: 苹果公司推出的一种便携式多功能数字多媒体播放器。

touch screen
触摸屏, 触屏

blurry /'ble:ri/
a. 模糊的,
难辨认的
convergence
/kən've:dʒəns/
n. 会聚; 结合;
收敛

microcontroller
/'maikrəukən'trəulə/
n. 微控制器
embedded
/im'bedid/
a. 嵌入(式)的

technicality
/tekni'kæliiti/
n. 技术性(细节)

relay /ri:'lei/
v. 接力传送; 传递;
转播
Web site
网站, 站点

software and their lack of a keyboard or touch screen puts severe limits on data entry.

Whether they fall into the handheld computer, smartphone, or portable media player categories, today's handheld digital devices have many features in common. The lines that separate these devices are blurry because the market is in a state of convergence. Currently, size and battery life seem to be the factors preventing a single device from integrating all the features possible for a handheld device. Technology, however, is likely to solve those problems soon.

VI. Microcontrollers

Many common appliances and machines are controlled by embedded microcontrollers. A microcontroller is a special-purpose microprocessor that is built into the machine it controls. A microcontroller is sometimes called a computer-on-a-chip or an embedded computer because it includes many of the elements common to computers.

A computer is defined as a multipurpose device that accepts input, produces output, stores data, and processes it according to a stored program. A microcontroller seems to fit the input, processing, output, and storage criteria that define computers. Some microcontrollers can even be reprogrammed to perform different tasks.

Technically, a microcontroller could be classified as a computer, just as smartphones and portable media players can be. Despite this technicality, however, microcontrollers tend to be referred to as processors rather than as computers because in practice they are used for dedicated applications, not as multipurpose devices.

Microcontrollers can be embedded in all sorts of everyday devices, enabling machines to perform sophisticated tasks that require awareness and feedback from the environment. When combined with wireless networks, devices with embedded processors can relay information to Web sites, cell phones, and a variety of data collection devices. Machines and appliances with embedded processors tend to be smarter about their use of resources—such as electricity and water—which makes them environmentally friendly.

Perhaps the most significant effect of microcontrollers is that they are an almost invisible technology, one that doesn't require much adaptation or

microcontrol

/ˈmaɪkrəʊkənˈtrəʊl/

v. 微控制

detrimental

/ˌdetrɪˈmentəl/

a. 有害的; 不利的

innocuous

/ɪˈnɒkjʊəs/

a. 无害的; 不得罪
(或冒犯) 人的**marketer** /ˈmɑːkɪtə/n. 在市场上做买卖
的人; 专营特定商
品的商人(或商号)**enforcement**

/ɪnˈfɔːsmənt/

n. 实施; 强制执行

learning on the part of the people who interact with microcontrolled devices. However, because microcontrollers remain mostly out-of-sight and out-of-mind, it is easy for their use to creep into areas that could be detrimental to quality of life, privacy, and freedom. That innocuous GPS chip in your cell phone, for example, can be useful if you're lost and need 911 assistance, but it could potentially be used by marketers, law enforcement¹, and others who want to track your location without your consent.

Exercises

I. Fill in the blanks with the information given in the text:

1. A computer accepts input, processes data, stores data, and produces _____ according to a series of instructions.
2. A(n) _____ specializes in compute-intensive problems.
3. A(n) _____ is a special-purpose microprocessor that is built into the machine it controls.
4. The lines that separate today's _____ digital devices are blurry because the market is in a state of _____.

II. Translate the following terms or phrases from English into Chinese and vice versa:

- | | |
|-----------------------------|-------------|
| 1. Web page | 5. 声卡 |
| 2. touch screen | 6. 存储服务器 |
| 3. blade server | 7. 便携式媒体播放器 |
| 4. removable storage device | 8. 硬盘 |

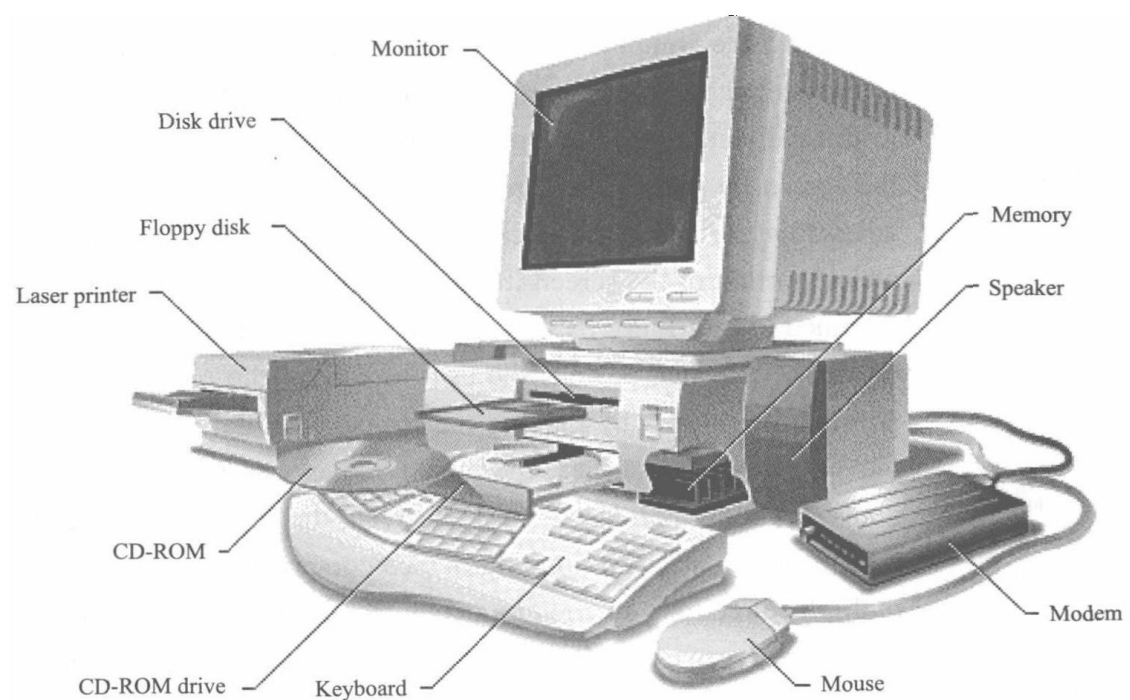
¹ law enforcement: 执法。这里指执法者。

Unit 2 Computer Architecture

(计算机体系结构)

Section A

Computer Hardware



I. Introduction

Computer hardware is the equipment involved in the function of a computer and consists of the components that can be physically handled. The function of these components is typically divided into three main categories: input, output, and storage. Components in these categories connect to microprocessors, specifically, the computer's central processing unit (CPU), the electronic circuitry that provides the computational ability and control of the computer, via wires or circuitry called a bus.

Software, on the other hand, is the set of instructions a computer uses to manipulate data, such as a word-processing program or a video game. These programs are usually stored and transferred via the computer's

video game
电子游戏

hardware to and from the CPU. Software also governs how the hardware is utilized; for example, how information is retrieved from a storage device. The interaction between the input and output hardware is controlled by software called the Basic Input/Output System (BIOS¹) software.

Although microprocessors are still technically considered to be hardware, portions of their function are also associated with computer software. Since microprocessors have both hardware and software aspects, they are therefore often referred to as **firmware**.

firmware

/ˈfɜːmwɛə/

n. [总称] 固件

II. Input Hardware

Input hardware consists of external devices—that is, components outside of the computer's CPU—that provide information and instructions to the computer. A **light pen** is a **stylus** with a light sensitive tip that is used to draw directly on a computer's video screen or to select information on the screen by pressing a clip in the light pen or by pressing the light pen against the surface of the screen. The pen contains light sensors that identify which portion of the screen it is passed over. A mouse is a pointing device designed to be gripped by one hand. It has a detection device (usually a ball) on the bottom that enables the user to control the motion of an on-screen **pointer**, or **cursor**, by moving the mouse on a flat surface. As the device moves across the surface, the cursor moves across the screen. To select items or choose commands on the screen, the user presses a button on the mouse. A joystick is a pointing device composed of a **lever** that moves in multiple directions to **navigate** a cursor or other graphical object on a computer screen. A keyboard is a typewriter-like device that allows the user to type in text and commands to the computer. Some keyboards have special **function keys** or integrated pointing devices, such as a **trackball** or touch-sensitive regions that let the user's finger motions move an on-screen cursor.

An optical **scanner** uses light-sensing equipment to convert images such as a picture or text into electronic signals that can be manipulated by a computer. For example, a photograph can be scanned into a computer and then included in a text document created on that computer. The two most common scanner types are the **flatbed scanner**, which is similar to an office **photocopier**, and the **hand-held scanner**, which is passed manually across the image to be processed. A microphone is a device for converting

light pen

光笔

stylus /ˈstajləs/

〔复〕-luses 或

-li /-lai/

n. 输入笔, 光笔

pointer /ˈpointə/

n. 指针(光标); 指示字, 指示符

cursor /ˈkɜːsə/

n. 光标

lever /ˈli:və; ˈle-/

n. (杠)杆; 控制杆

navigate /ˈneɪvɪgeɪt/

v. 航行(于); (为...)领航; 指引

function key

功能键, 操作键, 函数键

trackball /ˈtrækbɔːl/

n. 跟踪球, 轨迹球

scanner /ˈskænə/

n. 扫描仪;

扫描程序

flatbed scanner

平板扫描仪

photocopier

/ˈfəuteɪˈkɒpiə/

n. 复印机

hand-held scanner

手持式扫描仪

¹ BIOS: 基本输入/输出系统(Basic Input/Output System的首字母缩略), 读作 /ˈbaɪəʊs/。

module /'mɒdju:l/
n. 模块

modulator
/'mɒdjuleɪtə/
n. 调制器

demodulator
/di:'mɒdjuleɪtə/
n. 解调器

video display
视频显示 (器)

cathode /'kæθəʊd/
n. 阴极

cathode ray tube
阴极射线管

liquid crystal display
液晶显示 (器)

phosphorescent
/'fɒsfə'resənt/
a. 发磷光的

laptop /'læptɒp/
a. 膝上型的

matrix /'meɪtriks/
([复] -trices
/ -trɪsɪz / 或 -trɪks)
n. (矩) 阵

dot-matrix printer
点阵打印机

laser printer
激光打印机

drum /drʌm/
n. 磁鼓

toner /'təʊnə/
n. 色粉, 墨粉

fuse /fju:z/
v. 熔合; 熔凝

inkjet /'ɪŋkdʒet/
a. 喷墨的

inkjet printer
喷墨打印机

droplet /'drɒplɪt/
n. 小滴

retrieval /rɪ'tri:vəl/
n. 检索

drive /draɪv/
n. 驱动器

disk drive 磁盘机,
磁盘驱动器

floppy /'flɒpi/
a. (松) 软的

sound into signals that can then be stored, manipulated, and played back by the computer. A voice recognition **module** is a device that converts spoken words into information that the computer can recognize and process.

A modem, which stands for **modulator-demodulator**, is a device that connects a computer to a telephone line and allows information to be transmitted to or received from another computer. Each computer that sends or receives information must be connected to a modem. The information sent from one computer is converted by the modem into an audio signal, which is then transmitted by telephone lines to the receiving modem, which converts the signal into information that the receiving computer can understand.

III. Output Hardware

Output hardware consists of external devices that transfer information from the computer's CPU to the computer user. A **video display**, or screen, converts information generated by the computer into visual information. Displays commonly take one of two forms: a video screen with a **cathode ray tube** (CRT) or a video screen with a **liquid crystal display** (LCD). A CRT-based screen, or monitor, looks similar to a television set. Information from the CPU is displayed using a beam of electrons that scans a **phosphorescent** surface that emits light and creates images. An LCD-based screen displays visual information on a flatter and smaller screen than a CRT-based video monitor. LCDs are frequently used in **laptop** computers.

Printers take text and image from a computer and print them on paper. **Dot-matrix printers** use tiny wires to impact upon an inked ribbon to form characters. **Laser printers** employ beams of light to draw images on a **drum** that then picks up fine black particles called **toner**. The toner is fused to a page to produce an image. **Inkjet printers** fire **droplets** of ink onto a page to form characters and pictures.

IV. Storage Hardware

Storage hardware provides permanent storage of information and programs for **retrieval** by the computer. The two main types of storage devices are **disk drives** and memory. There are several types of disk drives: **hard**, **floppy**, **magneto-optical**, and **compact**. Hard disk drives store information in magnetic particles embedded in a disk. Usually a permanent part of the computer, hard disk drives can store large amounts of information and retrieve that information very quickly. **Floppy disk drives**

magneto-optical

/mæg.ni:təu'optikəl/

a. 磁光的

floppy disk

软(磁)盘

rigid /'ridʒid/

a. 硬的

compact disc

光盘

pit /pit/

n. 凹区, 凹槽

erase /'i:reiz/

v. 擦除, 消除, 清除

overwrite

/,əuve'reit/

v. 盖写, 改写, 重写

hard drive

硬盘驱动器

random /'rændəm/

a. 随机的; 任意的

random-access**memory**

随机(存取)

存储器

volatile /'vɒlətail/

a. 易失的; 易变的

volatile memory

易失性存储器

read-only memory

只读存储器

nonvolatile**memory**

非易失性存储器

also store information in magnetic particles embedded in removable disks that may be floppy or rigid. Floppy disks store less information than a hard disk drive and retrieve the information at a much slower rate. Magneto-optical disc¹ drives store information on removable discs that are sensitive to both laser light and magnetic fields. They can typically store as much information as hard disks but have slightly slower retrieval speeds. Compact disc drives (CD-ROM²) store information on pits burned into the surface of a disc of reflective material.³ The information stored on CD-ROMs cannot be erased or overwritten with new information. They can store about as much information as a hard drive but have a slower rate of information retrieval.

Memory refers to the computer chips that store information for quick retrieval by the CPU. Random-access memory (RAM⁴) is used to store the information and instructions that operate the computer's programs. Typically, programs are transferred from storage on a disk drive to RAM. RAM is also known as volatile memory because the information within the computer chips is lost when power to the computer is turned off. Read-only memory (ROM) contains critical information and software that must be permanently available for computer operation, such as the operating system that directs the computer's actions from start up to shut down. ROM is called nonvolatile memory because the memory chips do not lose their information when power to the computer is turned off.

Some devices serve more than one purpose. For example, floppy disks may also be used as input devices if they contain information to be used and processed by the computer user. In addition, they can be used as output devices if the user wants to store the results of computations on them.

V. Hardware Connections

To function, hardware requires physical connections that allow components to communicate and interact. A bus provides a common interconnected system composed of a group of wires or circuitry that coordinates and moves information between the internal parts of a computer.

¹ *disc*: *disc* 本来是英国英语的拼法, 其对应的美国拼法为 *disk*。但是, 在计算机英语中, 现在公认的标准做法是将 *disc* 用于表示光盘的场合, 而把 *disk* 用于所有其他的场合。因此, 本文在同一段落中出现了 *disc* 和 *disk* 两种拼写形式。另外, *optical disc* 与 *compact disc* 都指光盘。

² *CD-ROM*: 只读光盘(存储器)(*compact disc read-only memory* 的首字母缩略)。

³ *Compact disc drives (CD-ROM) store information on pits burned into the surface of a disc of reflective material.*: 由反射材料制成的光盘, 其表面烧蚀出凹陷区。光驱(CD-ROM)就是将信息存储于此。

⁴ *RAM*: 随机(存取)存储器(*random-access memory* 的首字母缩略), 读作 /ræm/。

address bus

地址总线

data bus

数据总线

A computer bus consists of two channels: one that the CPU uses to locate data, called the **address bus**, and another to send the data to that address, called the **data bus**. A bus is characterized by two features: how much information it can manipulate at one time, called the bus width, and how quickly it can transfer these data.

serial / 'sɪəriəl /

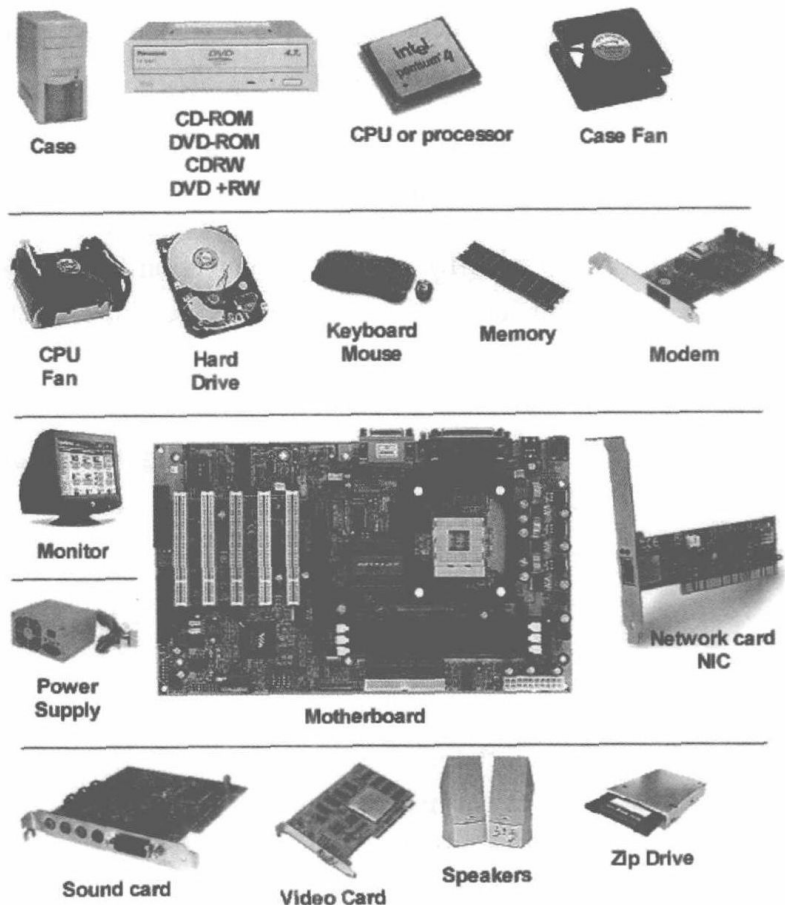
a. 串行的; 连续的

A **serial connection** is a wire or set of wires used to transfer information from the CPU to an external device such as a mouse, keyboard, modem, scanner, and some types of printers. This type of connection transfers only one piece of data at a time, and is therefore slow. The advantage of using a serial connection is that it provides effective connections over long distances.

block / blɒk /

n. (字、信息、程序、数据等的) 块; 分程序

A **parallel connection** uses multiple sets of wires to transfer blocks of information simultaneously. Most scanners and printers use this type of connection. A parallel connection is much faster than a serial connection, but it is limited to distances of less than 3 m (10 ft) between the CPU and the external device.

**motherboard**

/'mʌðəbɔ:d /

n. 母板, 主板

Figure 2A-1: Categories of Computer Hardware

Exercises

I. Fill in the blanks with the information given in the text:

1. The function of computer hardware is typically divided into three main categories. They are _____, _____, and _____.
2. The software that controls the interaction between the input and output hardware is called BIOS, which stands for _____.
3. The two most common types of scanners are _____ scanners and _____ scanners.
4. Video displays commonly take one of two forms. They are CRT-based screens and _____ screens.
5. In the text, the author mentions three of the most commonly used types of printers. They are _____ printers, laser printers, and _____ printers.
6. According to the text, the two main types of storage devices are _____ drives and _____.
7. It can be said that having a steady power source is important for a computer because RAM is _____ memory.
8. A(n) _____ connection transfers only one piece of data at a time while a(n) _____ connection transfers blocks of information at the same time.

II. Translate the following terms or phrases from English into Chinese and vice versa:

- | | |
|-----------------------------|---------------|
| 1. function key | 11. 操作系统 |
| 2. voice recognition module | 12. 液晶显示(器) |
| 3. touch-sensitive region | 13. 喷墨打印机 |
| 4. address bus | 14. 数据总线 |
| 5. flatbed scanner | 15. 串行连接 |
| 6. dot-matrix printer | 16. 易失性存储器 |
| 7. parallel connection | 17. 激光打印机 |
| 8. cathode ray tube | 18. 磁盘驱动器 |
| 9. video game | 19. 基本输入/输出系统 |
| 10. audio signal | 20. 视频显示器 |

III. Fill in each of the blanks with one of the words given in the following list, making changes if necessary:

<i>commercial</i>	<i>light</i>	<i>disc</i>	<i>optical</i>
<i>laser</i>	<i>represent</i>	<i>floppy</i>	<i>retrieve</i>
<i>access</i>	<i>store</i>	<i>erase</i>	<i>flat</i>
<i>reflect</i>	<i>equivalent</i>	<i>information</i>	<i>surface</i>

CD-ROM stands for compact disc read-only memory. Unlike _____ and hard disks, which use magnetic charges to represent 1s and 0s, _____ discs use reflected light. On a CD-ROM disc, 1s and 0s are represented by _____ areas and *bumpy* (高低不平的) areas (called "pits") on its bottom _____. The CD-ROM disc is read by a(n) _____ that projects a tiny beam of _____ on these areas. The amount of _____ light determines whether the area _____ a 1 or a 0.

Like a(n) _____ CD found in music stores, a CD-ROM is a "read-only" _____. Read-only means it cannot be written on or _____ by the user. Thus, you as a user have _____ only to the data *imprinted* (压印) by the publisher.

A single CD-ROM disc can _____ 650 *megabytes* (兆字节) of data. That is _____ to 451 floppy disks. With that much _____ on a single disc, the time to _____ or access the information is very important. An important characteristic of CD-ROM drives is their access rate.

IV. Translate the following passage from English into Chinese:

A modem is a device that converts between analog and digital signals. Digital signals, which are used by computers, are made up of separate units, usually represented by a series of 1's and 0's. Analog signals vary continuously; an example of an analog signal is a sound wave. Modems are often used to enable computers to communicate with each other across telephone lines. A modem converts the digital signals of the sending computer to analog signals that can be transmitted through telephone lines. When the signal reaches its destination, another modem reconstructs the original digital signal, which is processed by the receiving computer. If both modems can transmit data to each other simultaneously, the modems are operating in full *duplex* (双工的) mode; if only one modem can transmit at a time, the modems are operating in half duplex mode.

Section B

Components of an Operating System



shell /ʃel/

n. 外壳; 命令解释程序; 对象处理程序

graphical user interface

图形用户界面

pictorial /pik'tɔ:riəl/

a. 用图表示的, 图示的; 形象化的

icon /'aɪkən/

n. 图标, 图符

textual

/ˈtekstʃuəl, -tʃuəl/

a. 文本的, 正文的

functionality

/ˌfʌŋkʃə'næləti/

n. 功能性

I. The Shell of an Operating System

In order to perform the actions requested by the computer's users, an operating system must be able to communicate with those users. The portion of an operating system that handles this communication is often called the shell. Modern shells perform this task by means of a **graphical user interface** (GUI¹—pronounced “GOO-ee”) in which objects to be manipulated, such as files and programs, are represented **pictorially** on the monitor screen as **icons**. These systems allow users to issue commands to the operating system by pointing to these icons by means of a mouse and pressing a button on the mouse. Older shells communicate with users through **textual** messages using a keyboard and monitor screen.

Although an operating system's shell plays an important role in establishing a machine's **functionality**, this shell is merely an interface

¹ GUI: 图形用户界面 (graphical user interface 的首字母缩略), 读作 /'gu:ɪ/。

compatible

/kəm'pætəbl/

a. 兼容的

manager

/'mænidʒə/

n. 管理程序,
管理器**allocate** /'æləkeɪt/

v. 分配; 分派

click /klik/

v. (鼠标) 单击

between a user and the real heart of the operating system (Figure 2B-1). This distinction between the shell and the internal parts of the operating system is emphasized by the fact that some operating systems allow a user to select among different shells to obtain the most **compatible** interface for that particular user. Users of the UNIX¹ operating system, for example, can select among a variety of shells including the Bourne shell, the C shell, and the Korn shell.

An important component within today's GUI shells is the window **manager**, which **allocates** blocks of space on the screen, called windows, and keeps track of which application is associated with each window. When an application wants to display something on the screen, it notifies the window manager, and the window manager places the desired image in the window assigned to the application. In turn, when a mouse button is **clicked**, it is the window manager that computes the mouse's location on the screen and notifies the appropriate application of the mouse action.

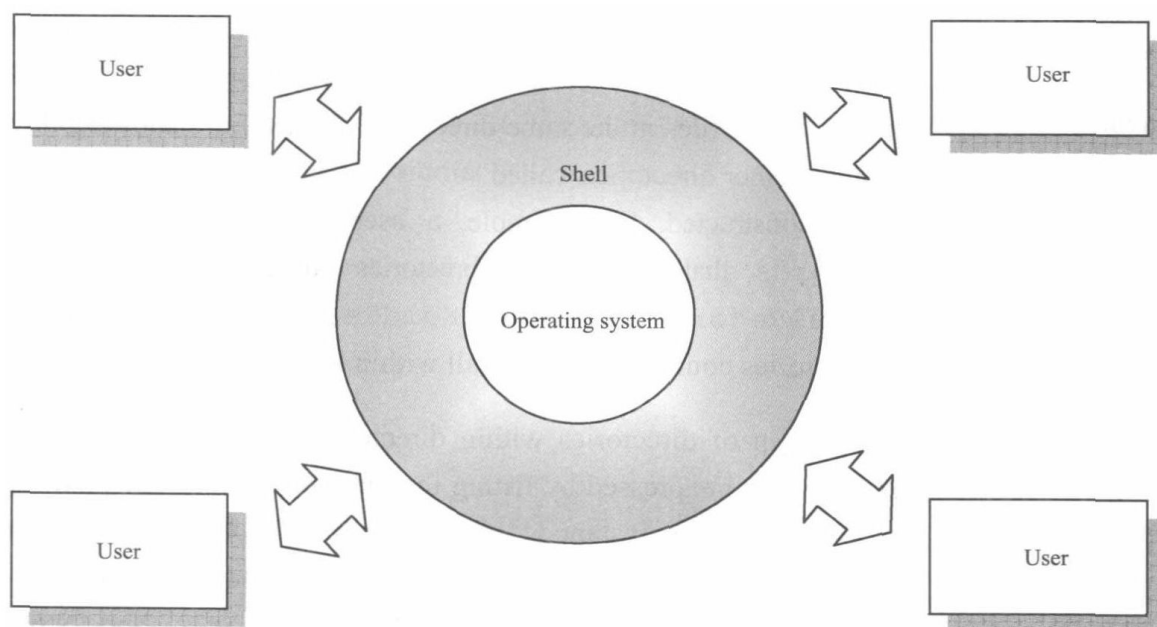


Figure 2B-1: The Shell as an Interface between Users and the Operating System

¹ **UNIX**: UNIX (读作 /'ju:niks/) 操作系统, 系 AT&T 公司 (美国电话电报公司) 开发的一种操作系统, 1971 年首次运行在 PDP-11 (美国 DEC 公司, 即数字设备公司, 推出的一种小型机) 上, 主要开发人为 肯·汤普森 (Ken Thompson, 1943—) 与丹尼斯·里奇 (Dennis Ritchie, 1941—2011)。

kernel /'kə:nəl/n. 内核, 内核程序,
核心程序

II. The Kernel of an Operating System

In contrast to an operating system's shell, the internal part of an operating system is called the kernel. An operating system's kernel contains those software components that perform the very basic functions required by the computer installation.

1. File Manager

mass storage海量存储器,
大容量存储器**extension**

/ik'stenʃən/

n. 扩展, 扩充;
扩展名**directory**

/di'rektəri/

n. 目录

folder /'fəʊldə/

n. 文件夹

subdirectory

/sʌbdɪ'rektəri/

n. 子目录

hierarchical

/haɪə'rɔ:kɪkəl/

a. 分级的, 分层的,
层次的**directory path**

目录路径

slash /slæʃ/

n. 斜线, 斜杠

prehistoric

/pri:'hɪstɒrɪk/

a. 史前的

dinosaur /'daɪnəsɔ:/n. 恐龙; (尤指废弃
过时的) 庞然大物**terminate**

/'tɜ:mɪneɪt/

v. (使) 终止

path expression

路径表达式

One such unit is the file manager, whose job is to coordinate the use of the machine's mass storage facilities. More precisely, the file manager maintains records of all the files stored in mass storage, including where each file is located, which users are allowed to access the various files, and which portions of mass storage are available for new files or extensions to existing files. These records are kept on the individual storage medium containing the related files so that each time the medium is placed online, the file manager can retrieve them and thus know what is stored on that particular medium.

For the convenience of the machine's users, most file managers allow files to be grouped into a bundle called a directory or folder. This approach allows a user to organize his or her files according to their purposes by placing related files in the same directory. Moreover, by allowing directories to contain other directories, called subdirectories, a hierarchical organization can be constructed. For example, a user may create a directory called MyRecords that contains subdirectories called FinancialRecords, MedicalRecords, and HouseHoldRecords. Within each of these subdirectories could be files that fall within that particular category.

A chain of directories within directories is called a directory path. Paths are often expressed by listing the directories along the path separated by slashes. For instance, animals/prehistoric/dinosaurs would represent the path starting at the directory named animals, passing through its subdirectory named prehistoric, and terminating in the sub-subdirectory dinosaurs. (For Windows users the slashes in such a path expression are reversed as in animals\prehistoric\dinosaurs.)

Any access to a file by other software units is obtained at the

discretion

/dis'kreʃən/

n. 斟酌决定 (或处理) 的自由

main memory

主存 (存储器)

descriptor

/di'skriptə/

n. 描述符, 解说符

reference

/'refərəns/

v. 引用, 参考

driver /'draɪvə/

n. 驱动程序, 驱动器

device driver

设备驱动程序

controller

/kən'trəʊlə/

n. 控制器

peripheral

/pə'rɪfərəl/

a. 外围的, 外部的

peripheral device

外围设备, 外部设备

generic /dʒɪ'nerɪk/

a. 类属的; 一般的; 通用的

decode /di:'keud/

v. 译 (码), 解 (码)

status word

状态字

handshaking

/'hændʃeɪkɪŋ/

n. 握手, 信号交换

customize

/'kʌstəmaɪz/

v. 定制, 使用户化

minimal /'mɪnɪməl/

a. 最小的; 最低限度的

multitasking

/'mʌltɪtɑːskɪŋ/

n. 多任务 (处理)

discretion of¹ the file manager. The procedure begins by requesting that the file manager grant access to the file through a procedure known as opening the file. If the file manager approves the requested access, it provides the information needed to find and to manipulate the file. This information is stored in an area of **main memory** called a file **descriptor**. It is by **referencing** the information in this file descriptor that individual operations are performed on the file.

2. Device Driver

Another component of the kernel consists of a collection of **device drivers**, which are the software units that communicate with the **controllers** (or at times, directly with **peripheral devices**) to carry out operations on the peripheral devices attached to the machine. Each device driver is uniquely designed for its particular type of device (such as a printer, disk drive, or monitor) and translates **generic** requests into the more technical steps required by the device assigned to that driver. For example, a device driver for a printer contains the software for reading and **decoding** that particular printer's **status word** as well as all the other **handshaking** details. Thus, other software components do not have to deal with those technicalities in order to print a file. Instead, the other components can merely rely on the device driver software to print the file, and let the device driver take care of the details. In this manner, the design of the other software units can be independent of the unique characteristics of particular devices. The result is a generic operating system that can be **customized** for particular peripheral devices by merely installing the appropriate device drivers.

3. Memory Manager

Still another component of an operating system's kernel is the memory manager, which is charged with the task of coordinating the machine's use of main memory. Such duties are **minimal** in an environment in which a computer is asked to perform only one task at a time. In these cases, the program for performing the current task is placed at a predetermined location in main memory, executed, and then replaced by the program for performing the next task. However, in multiuser or **multitasking** environments in which the computer is asked to address many needs at the same time, the duties of the memory manager are extensive. In these cases,

¹ at the discretion of: 由...斟酌决定; 随...的意见。

reside /ri'zaid/

v. 驻留, 常住

concurrent

/kən'kʌrənt/

a. 同时发生的, 并发的, 并行的

allot /ə'lɒt/

v. 分配; 分派

paging /'peɪdʒɪŋ/

n. 页面调度(技术或方法), 分页

magnetic disk

磁盘

bit pattern

位模式

shuffle /'ʃʌfl/

v. 混洗, 洗牌(并行处理中的一种操作)

fictional /'fɪkʃənəl/

a. 小说的; 虚构的

virtual memory

虚(拟)存(储器), 虚拟存储(器), 虚拟内存

scheduler

/'ʃedju:lə; 'ske-

dʒu:lə/

n. 调度程序

dispatcher

/dɪs'pætʃə/

n. 调度程序

time-sharing

/'taɪm,ʃeərɪŋ/

n. & a. 分时(技术或方法)(的)

many programs and blocks of data must **reside** in main memory **concurrently**. Thus, the memory manager must find and assign memory space for these needs and ensure that the actions of each program are restricted to the program's **allotted** space. Moreover, as the needs of different activities come and go, the memory manager must keep track of those memory areas no longer occupied.

The task of the memory manager is complicated further when the total main memory space required exceeds the space actually available in the computer. In this case, the memory manager may create the illusion of additional memory space by rotating programs and data back and forth between main memory and mass storage (a technique called **paging**). Suppose, for example, that a main memory of 1024MB¹ is required but the computer only has 512MB. To create the illusion of the larger memory space, the memory manager reserves 1024MB of storage space on a **magnetic disk**. There it records the **bit patterns** that would be stored in main memory if main memory had an actual capacity of 1024MB. This data is divided into uniform sized units called pages, which are typically a few KB² in size. Then the memory manager **shuffles** these pages back and forth between main memory and mass storage so that the pages that are needed at any given time are actually present in the 512MB of main memory. The result is that the computer is able to function as though it actually had 1024MB of main memory. This large "fictional" memory space created by **paging** is called **virtual memory**.

4. Scheduler and Dispatcher

One of the most fundamental concepts of modern operating systems is the distinction between a program and the activity of executing a program. The former is a static set of directions, whereas the latter is a dynamic activity whose properties change as time progresses. This activity is known as a process. In a typical **time-sharing** computer installation, many processes are normally competing for the computer's resources. The tasks associated with coordinating the execution of processes are handled by the scheduler and dispatcher within the operating system's kernel.

The scheduler maintains a record of the processes present in the computer system, introduces new processes to this pool, and removes

¹ MB: 兆字节 (megabyte 的缩略)。

² KB: 千字节 (kilobyte 的缩略)。

process table
进程表

time slice

时间片

millisecond

/ˈmɪl.i.sek.ənd/

n. 毫秒

process switch

进程转换

context switch

上下文转换,
语境转换

initiate /ɪˈnɪʃieɪt/

v. 开始; 发起

timer /ˈtaɪmə/

n. 计时器, 定时器;
定时程序

interrupt

/ˌɪntəˈrʌpt/

n. 中断

interrupt signal

中断信号

handler /ˈhændlə/

n. 处理程序,
处理器

interrupt handler

中断处理程序

preempt

/ˌpriːˈempt, pri-/

v. 预先制止; 抢先,
先占, 预占

update /ʌpˈdeɪt/

v. 更新; 修改

completed processes from the pool. Thus when a user requests the execution of an application, it is the scheduler that adds the execution of that application to the pool of current processes. To keep track of all the processes, the scheduler maintains a block of information in main memory called the **process table**. Each time the execution of a program is requested, the scheduler creates a new entry for that process in the process table. This entry contains such information as the memory area assigned to the process, the priority of the process, and whether the process is ready or waiting. A process is ready if it is in a state in which its progress can continue; it is waiting if its progress is currently delayed until some external event occurs, such as the completion of a disk access, the pressing of a key at the keyboard, or the arrival of a message from another process.

The dispatcher is the component of the kernel that ensures that the scheduled processes are actually executed. In a time-sharing system, this task is accomplished by time-sharing; that is, dividing time into short segments, each called a **time slice** (typically no more than 50 **milliseconds**), and then switching the CPU's attention among the processes as each is allowed to execute for one time slice. The procedure of changing from one process to another is called a **process switch** (or a **context switch**). Each time the dispatcher awards a time slice to a process, it **initiates** a **timer** circuit that will indicate the end of the slice by generating a signal called an **interrupt**. When the CPU receives an **interrupt signal**, it completes its current machine cycle, saves its position in the current process and begins executing a program, called an **interrupt handler**, which is stored at a predetermined location in main memory. This interrupt handler is a part of the dispatcher, and it describes how the dispatcher should respond to the interrupt signal. Thus, the effect of the interrupt signal is to **preempt** the current process and transfer control back to the dispatcher. At this point, the dispatcher first allows the scheduler to **update** the process table (for instance, the priority of the process that has just completed its time slice may need to be lowered and the priorities of other processes may need to be raised). Then, the dispatcher selects the process from the process table that has the highest priority among the ready processes, restarts the timer circuit, and allows the selected process to begin its time slice.

Exercises

I. Fill in the blanks with the information given in the text:

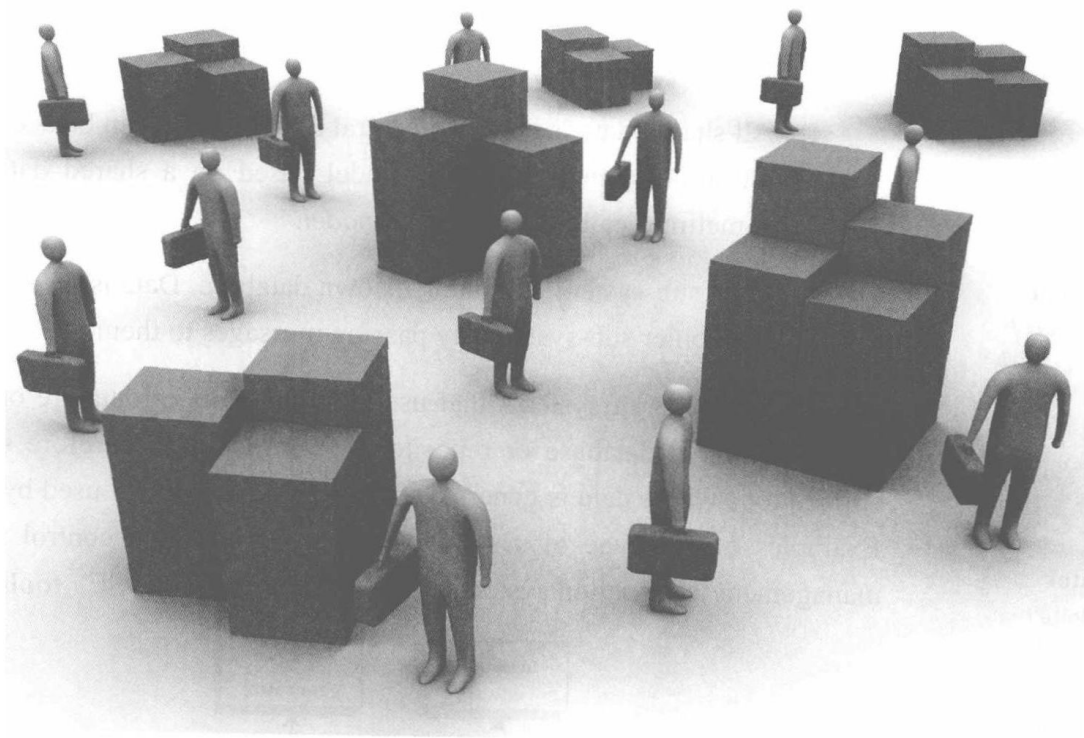
1. Modern shells perform the task of communicating with computer users by means of a _____ user interface.
2. The kernel of a typical operating system contains such software components as the _____ manager, various device drivers, the memory manager, the _____, and the dispatcher.
3. When the total main memory space required exceeds the space actually available in the computer, the memory manager may resort to _____ memory.
4. In a time-sharing system, time is divided into short segments, each of which is called a time _____.

II. Translate the following terms or phrases from English into Chinese and vice versa:

- | | |
|----------------------|-----------|
| 1. interrupt handler | 6. 外围设备 |
| 2. virtual memory | 7. 进程表 |
| 3. context switch | 8. 时间片 |
| 4. main memory | 9. 图形用户界面 |
| 5. bit pattern | 10. 海量存储器 |

Section C

System Organization



I. Introduction

The organization of a system reflects the basic strategy that is used to structure a system. You have to make decisions on the overall organizational model of a system early in the architectural design process. The system organization may be directly reflected in the sub-system structure. However, it is often the case that the sub-system model includes more detail than the organizational model, and there is not always a simple **mapping** from sub-systems to organizational structure¹.

This section discusses three organizational styles that are very widely used. These are a shared data **repository** style, a shared services and servers style and an **abstract machine** or **layered** style where the system is

map /mæp/
v. 映射; 变换,
变址

repository
/ri'pɒzɪtəri/
n. 存储库, 仓库

abstract machine
抽象机

layered /'leɪəd/
a. 分层的

¹ *there is not always a simple mapping from sub-systems to organizational structure*: 从子系统到组织结构并非总是一种简单的映射。这是一个部分否定的结构。

tier / tie /
n. (一) 层

organized as a **tier** of functional layers. These styles can be used separately or together. For example, a system may be organized around a shared data repository but may construct layers around this to present a more abstract view of the data.

II. The Repository Model

Sub-systems making up a system must exchange information so that they can work together effectively. There are two fundamental ways in which this can be done.

1. All shared data is held in a central database that can be accessed by all sub-systems. A system model based on a shared database is sometimes called a *repository model*.
2. Each sub-system maintains its own database. Data is **interchanged** with other sub-systems by passing messages to them.

interchange
/ .intə'tʃeindʒ /
v. 交换, 互换

The majority of systems that use large amounts of data are organized around a shared database or repository. This model is therefore suited to applications where data is generated by one sub-system and used by another. Examples of this type of system include command and control systems, management information systems, CAD¹ systems and CASE² toolsets.

toolset / 'tu:lset /
n. 成套工具, 工具箱
translator
/ trəns'leɪtə /
n. 翻译程序, 翻译器
editor / 'edɪtə /
n. 编辑程序, 编辑器
generator
/ 'dʒenəreɪtə /
n. 生成程序, 生成器
analyzer / 'ænəlaɪzə /
n. 分析程序, 分析算法, 分析器

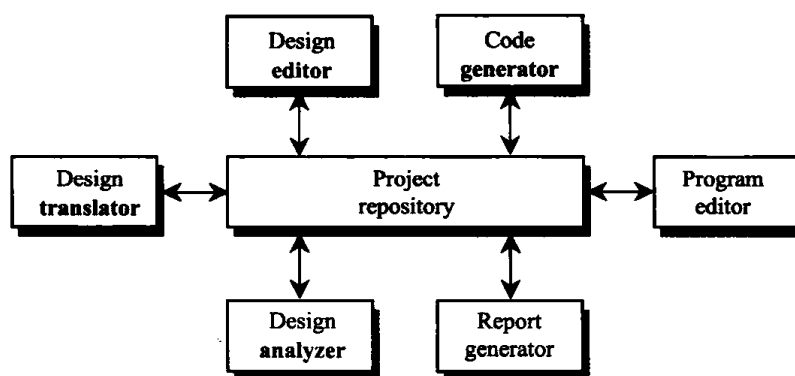


Figure 2C-1: The Architecture of an Integrated CASE Toolset

Figure 2C-1 is an example of a CASE toolset architecture based on a shared repository. The first shared repository for CASE tools was probably developed in the early 1970s by a UK company called ICL³ to support their

¹ CAD: 计算机辅助设计 (computer-aided design 的首字母缩略)。

² CASE: 计算机辅助软件工程 (computer-aided software engineering 的首字母缩略)。

³ ICL: 国际计算机有限公司 (International Computers Limited 的首字母缩略)。英国的一家大型计算机硬件、软件及服务公司, 成立于 1968 年, 1981 年被日本富士通株式会社兼并, 2002 年更名为 Fujitsu Services Limited (富士通服务有限公司)。

operating system development. This model became more widely known when Buxton¹ made proposals to support the development of systems written in Ada². Since then, many CASE toolsets have been developed around a shared repository.

The advantages and disadvantages of a shared repository are as follows:

1. It is an efficient way to share large amounts of data. There is no need to transmit data explicitly from one sub-system to another.
2. However, sub-systems must agree on the repository data model. Inevitably, this is a compromise between the specific needs of each tool. Performance may be adversely affected by this compromise. It may be difficult or impossible to integrate new sub-systems if their data models do not fit the agreed schema.
3. Sub-systems that produce data need not be concerned with how that data is used by other sub-systems.
4. However, evolution may be difficult as a large volume of information is generated according to an agreed data model. Translating this to a new model will certainly be expensive; it may be difficult or even impossible.
5. Activities such as backup, security, access control and recovery from error are centralized. They are the responsibility of the repository manager. Tools can focus on their principal function rather than be concerned with these issues.
6. However, different sub-systems may have different requirements for security, recovery and backup policies. The repository model forces the same policy on all sub-systems.
7. The model of sharing is visible through the repository schema. It is straightforward to integrate new tools given that they are compatible with the agreed data model.³

adverse

/ 'ædvə:s, æd'və:s /

a. 不利的, 有害的

schema / 'ski:mə /

n. 模式; 图表; 纲要

backup / 'bækʌp /

n. 备份, 后备

straightforward

/ ,streɪt'fɔ:wəd /

a. 径直的; 正直的;
简单的

¹ Buxton: 即 John N. Buxton, 约翰 N. 巴克斯顿, 英国计算机科学家。

² Ada: Ada 语言。

³ It is straightforward to integrate new tools given that they are compatible with the agreed data model.: 只要新工具与商定的数据模型兼容, 将新工具结合进去就会很容易。句中的 given 为介词, given 作介词时后面可跟 that 从句, 表示“考虑到”、“假设”、“倘若”等意思。

redundancy

/ri'dʌndənsi/

n. 冗余

inconsistency

/ˌɪnkən'sistənsi/

n. 不一致

blackboard model

黑板法模型

trigger /'trɪɡə/

v. 触发

activate /'æktiveɪt/

v. 激活, 启动

8. However, it may be difficult to distribute the repository over a number of machines. Although it is possible to distribute a logically centralized repository, there may be problems with data **redundancy** and **inconsistency**.

In the above model, the repository is passive and control is the responsibility of the sub-systems using the repository. An alternative approach has been derived for AI systems that use a “**blackboard**” model, which **triggers** sub-systems when particular data become available. This is appropriate when the form of the repository data is less well structured. Decisions about which tool to **activate** can only be made when the data has been analyzed.

III. The Client-Server Model

The client-server architectural model is a system model where the system is organized as a set of services and associated servers and clients that access and use the services. The major components of this model are:

1. A set of servers that offer services to other sub-systems. Examples of servers are print servers that offer printing services, file servers that offer file management services and a compile server, which offers programming language **compilation** services.
2. A set of clients that call on the services offered by servers. These are normally sub-systems in their own right.¹ There may be several instances of a client program executing concurrently.
3. A network that allows the clients to access these services. This is not strictly necessary as both the clients and the servers could run on a single machine. In practice, however, most client-server systems are implemented as **distributed** systems.

compilation

/ˌkɒmpɪˈleɪʃən/

n. 编译; 汇编

distributed

/dɪ'strɪbʊtɪd/

a. 分布(式)的

procedure call

过程调用

protocol

/'prəʊtəˌkɒl/

n. 协议

Clients may have to know the names of the available servers and the services that they provide. However, servers need not know either the identity of clients or how many clients there are. Clients access the services provided by a server through remote **procedure calls** using a request-reply **protocol** such as the http² protocol used in the WWW³. Essentially, a client makes a request to a server and waits until it receives a reply.

¹ *These are normally sub-systems in their own right.*: 这些(客户机)通常本身就是子系统。短语 in one's own right 表示“凭本身的权利(或能力、实力、资格等)”。

² *http*: 亦作 HTTP, 超(级)文本传送(或传输)协议(Hypertext Transfer Protocol 的缩略)。

³ *WWW*: 万维网(World Wide Web 的首字母缩略)。

library /'laibrəri/
n. 库, 程序 (或文件、对象) 库

frame /freim/
n. 帧, 画面

synchrony
/'sɪŋkrəni/
n. 同时 (性); 同步 (性)

store /stɔ:/
n. 存储 (器)

compression
/kəm'preʃən/
n. 压缩

decompression
/di:kəm'preʃən/
n. 解压缩

clip /klɪp/
n. 剪下来的东西; 电影 (或电视) 片段

film clip
影片

digitize /'dɪdʒɪtaɪz/
v. 使 (数据) 数字化

info /'ɪnfəʊ/
n. 〈口〉信息
(= information)

query /'kwɪəri/
n. 查询

integrated user interface
综合用户接口

browser /'braʊzə/
n. 浏览器

Web browser
网络浏览器

network /'netwɜ:k/
v. 联网

upgrade
/ʌp'greɪd, 'ʌpg-/
v. 使升级; 改善

Figure 2C-2 shows an example of a system that is based on the client-server model. This is a multi-user, Web-based system to provide a film and photograph library.¹ In this system, several servers manage and display the different types of media. Video frames need to be transmitted quickly and in synchrony but at relatively low resolution. They may be compressed in a store, so the video server may handle video compression and decompression into different formats. Still pictures, however, must be maintained at a high resolution, so it is appropriate to maintain them on a separate server.

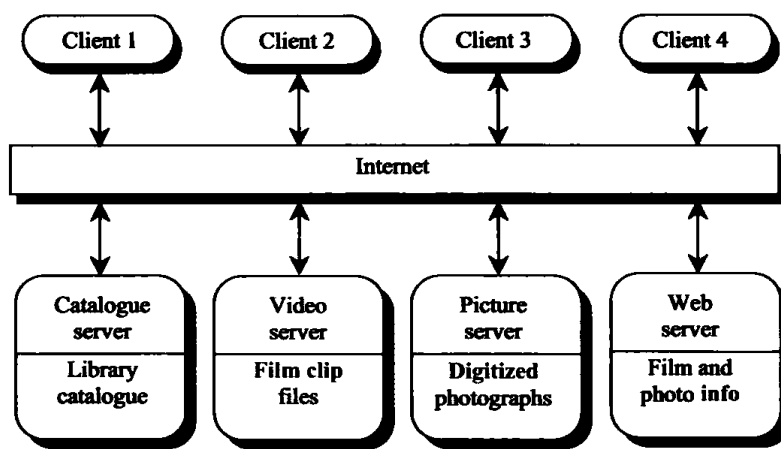


Figure 2C-2: The Architecture of a Film and Picture Library System

The catalogue must be able to deal with a variety of queries and provide links into the Web information system that includes data about the film and video clip, and an e-commerce system that supports the sale of film and video clips. The client program is simply an integrated user interface, constructed using a Web browser, to these services.²

The most important advantage of the client-server model is that it is a distributed architecture. Effective use can be made of networked systems with many distributed processors. It is easy to add a new server and integrate it with the rest of the system or to upgrade servers transparently without affecting other parts of the system.

However, changes to existing clients and servers may be required to gain the full benefits of integrating a new server. There may be no shared

¹ *This is a multi-user, Web-based system to provide a film and photograph library.*: 这是一个多用户的网基系统, 用于提供一个电影与照片库。

² *The client program is simply an integrated user interface, constructed using a Web browser, to these services.*: 客户程序只是一个用网络浏览器构建的连接这些服务的综合用户接口。interface 后面跟有两个定语: constructed using a Web browser 和 to these services。

optimize /'ɒptimaɪz/
v. 使优化,
使最佳化

data model across servers and sub-systems may organize their data in different ways. This means that specific data models may be established on each server to allow its performance to be **optimized**. Of course, if an XML¹-based representation of data is used, it may be relatively simple to convert from one schema to another. However, XML is an inefficient way to represent data, so performance problems can arise if this is used.

IV. The Layered Model

The layered model of an architecture (sometimes called an abstract machine model) organizes a system into layers, each of which provide a set of services. Each layer can be thought of as an abstract machine whose machine language is defined by the services provided by the layer. This “language” is used to implement the next level of abstract machine. For example, a common way to implement a language is to define an ideal “language machine” and compile the language into code for this machine. A further translation step then converts this abstract machine code to real machine code.

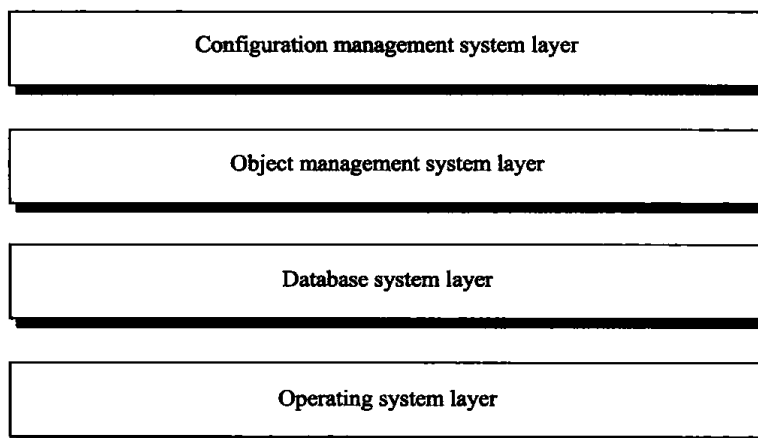


Figure 2C-3: The Layered Model of a Version Management System

An example of a layered model is the OSI² reference model of network protocols. Another influential example was proposed by Buxton, who suggested a three-layer model for an Ada Programming Support Environment (APSE). Figure 2C-3 reflects the APSE structure and shows how a configuration management system might be integrated using this abstract machine approach.

¹ XML: 可扩展标记语言 (eXtensible Markup Language 的缩略)。

² OSI: 开放式系统互连 (Open Systems Interconnection 的首字母缩略)。OSI 模型是由国际标准化组织 (ISO) 设计的。该模型理论上允许任意两个不同系统 (如计算机) 互相通信而无须考虑它们的基本体系结构。

configuration item

配置项

rollback /'rəʊlbæk/

n. 回滚, 回退; 重新运行, 重算

underlying

/ˌʌndə'laɪɪŋ/

a. 基本的, 根本的

filestore /'faɪlstɔː/

n. 文件存储(器)

incremental

/ˌɪnkri'mentəl/

a. 增量的; 递增的

adjacent

/ə'dʒeɪsənt/

a. 相邻的, 毗连的

localize /'ləʊkəlaɪz/

v. 使局部化, 使本地化

dependency

/di'pendənsi/

n. 依靠, 依赖

subvert

/səb've:t, sʌb-/

v. 颠覆; (暗中)

破坏; 搅乱

predecessor

/ˈpriːdisesə; 'pre-/

n. 前任; 前身;

先驱, 先驱

The configuration management system manages versions of objects and provides general configuration management facilities. To support these configuration management facilities, it uses an object management system that provides information storage and management services for configuration items or objects. This system is built on top of a database system to provide basic data storage and services such as transaction management, rollback and recovery, and access control. The database management uses the underlying operating system facilities and filestore in its implementation.

The layered approach supports the incremental development of systems. As a layer is developed, some of the services provided by that layer may be made available to users. This architecture is also changeable and portable. So long as its interface is unchanged, a layer can be replaced by another, equivalent layer. Furthermore, when layer interfaces change or new facilities are added to a layer, only the adjacent layer is affected. As layered systems localize machine dependencies in inner layers, this makes it easier to provide multi-platform implementations of an application system. Only the inner, machine-dependent layers need be reimplemented to take account of the facilities of a different operating system or database.

A disadvantage of the layered approach is that structuring systems in this way can be difficult. Inner layers may provide basic facilities, such as file management, that are required at all levels. Services required by a user of the top level may therefore have to “punch through” adjacent layers to get access to services that are provided several levels beneath it. This subverts the model, as the outer layer in the system does not just depend on its immediate predecessor.

Performance can also be a problem because of the multiple levels of command interpretation that are sometimes required. If there are many layers, a service request from a top layer may have to be interpreted several times in different layers before it is processed. To avoid these problems, applications may have to communicate directly with inner layers rather than use the services provided by the adjacent layer.

Exercises

I. Fill in the blanks with the information given in the text:

1. The three widely used organizational models for systems discussed in the text are the _____ model, client-server model, and layered model.
2. There are two fundamental ways in which sub-systems making up a system exchange information: all shared data is held in a(n) _____ database accessible to all of them; and each maintains its own database and interchanges data with other _____.
3. The major components of the client-server model usually include a set of servers, a set of clients, and a(n) _____.
4. The OSI reference model of network protocols is an example of the _____ model.

II. Translate the following terms or phrases from English into Chinese and vice versa:

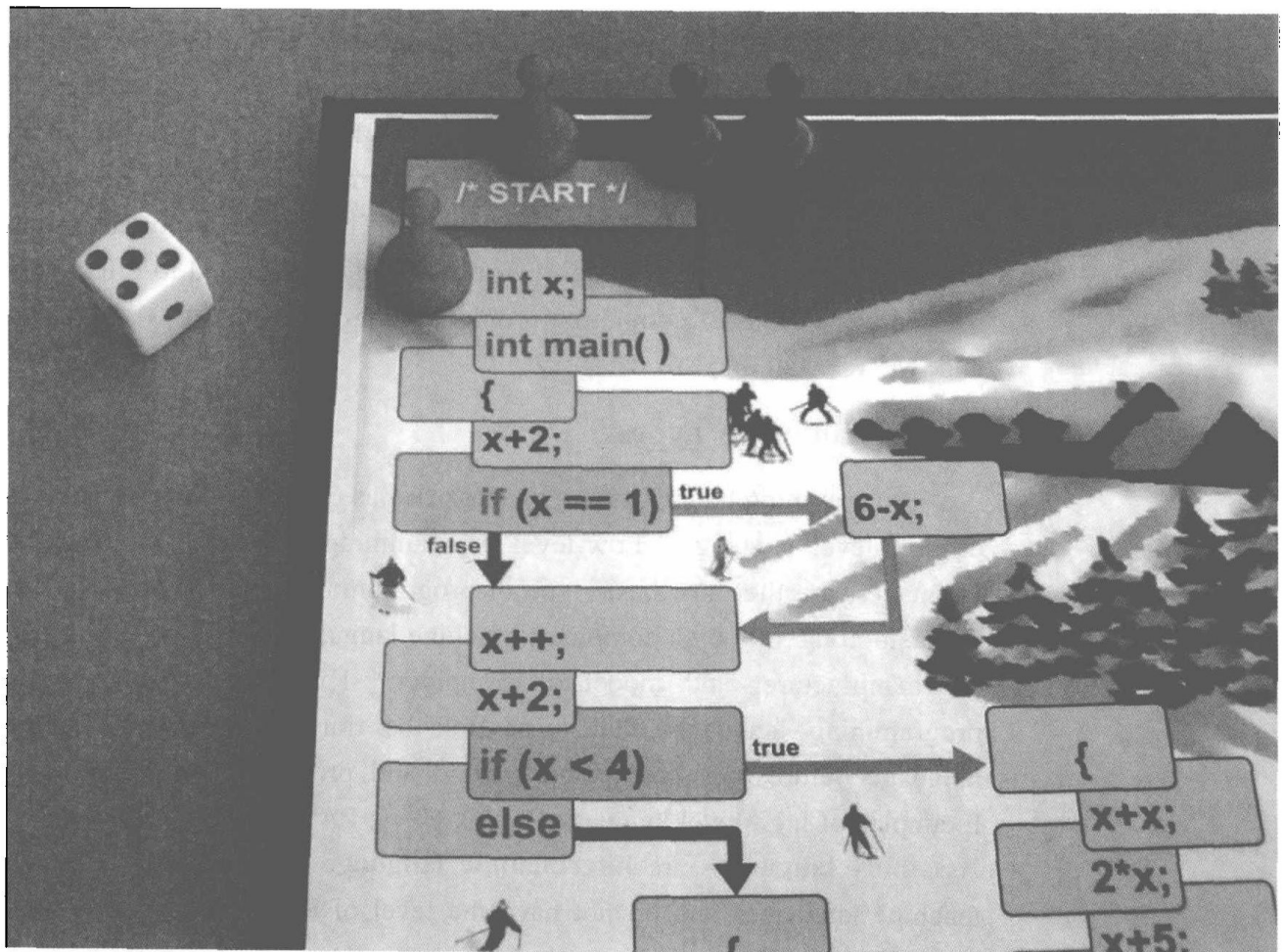
- | | |
|-----------------------|-------------|
| 1. code generator | 5. 计算机辅助设计 |
| 2. abstract machine | 6. 数据冗余 |
| 3. program editor | 7. 指挥与控制系统 |
| 4. configuration item | 8. 视频压缩与解压缩 |

Unit 3 Computer Language and Programming

(计算机语言与编程)

Section A

Programming Language



syntax /'sintæks/

n. 句法

ambiguous

/æm'bigjuəs/

a. 含糊不清的,
模棱两可的

grammatical

/gre'mætɪkəl/

a. (符合) 语法的

I. Introduction

Programming languages, in computer science, are the artificial languages used to write a sequence of instructions (a computer program) that can be run by a computer. Similar to natural languages, such as English, programming languages have a vocabulary, grammar, and **syntax**. However, natural languages are not suited for programming computers because they are **ambiguous**, meaning that their vocabulary and **grammatical** structure

punctuation

/ˌpʌŋktʃu'eɪʃən/

n. 标点符号

sophistication

/səˌfɪstɪ'keɪʃən/

n. 复杂性; 尖端性

may be interpreted in multiple ways. The languages used to program computers must have simple logical structures, and the rules for their grammar, spelling, and punctuation must be precise.

Programming languages vary greatly in their sophistication and in their degree of versatility. Some programming languages are written to address a particular kind of computing problem or for use on a particular model of computer system. For instance, programming languages such as FORTRAN¹ and COBOL² were written to solve certain general types of programming problems—FORTRAN for scientific applications, and COBOL for business applications. Although these languages were designed to address specific categories of computer problems, they are highly portable, meaning that they may be used to program many types of computers. Other languages, such as machine languages, are designed to be used by one specific model of computer system, or even by one specific computer in certain research applications. The most commonly used programming languages are highly portable and can be used to effectively solve diverse types of computing problems. Languages like C³, PASCAL⁴, and BASIC⁵ fall into this category.

II. Language Types

Programming languages can be classified as either low-level languages or high-level languages. Low-level programming languages, or machine languages, are the most basic type of programming languages and can be understood directly by a computer. Machine languages differ depending on the manufacturer and model of computer.⁶ High-level languages are programming languages that must first be translated into a machine language before they can be understood and processed by a computer. Examples of high-level languages are C, C++⁷, PASCAL, and FORTRAN. Assembly languages are intermediate languages that are very close to machine languages and do not have the level of linguistic sophistication

assembly language

汇编语言

intermediate

language

中间语言, 中级语言

linguistic

/lɪŋ'ɡwɪstɪk/

a. 语言(学)的

¹ **FORTRAN**: FORTRAN 语言, 公式翻译程序语言 (Formula Translation 的缩合), 读作 /'fɔ:træn/。

² **COBOL**: COBOL 语言, 面向商业的通用语言 (Common Business-Oriented Language 的缩略), 读作 /'kəʊbɒl/。

³ **C**: C 语言, 一种程序设计语言。

⁴ **PASCAL**: PASCAL 语言, 为纪念法国数学家和哲学家布莱斯·帕斯卡 (Blaise Pascal) 而命名, 读作 /'pæskəl/ 或 /pɑ:s'kɑ:l/。

⁵ **BASIC**: BASIC 语言, 初学者通用符号指令码 (Beginner's All-purpose Symbolic Instruction Code 的首字母缩略), 读作 /'beɪsɪk/。

⁶ *Machine languages differ depending on the manufacturer and model of computer.*: 机器语言视计算机制造商与型号不同而有所区别。

⁷ **C++**: C++ 语言, 一种面向对象的程序设计语言。

exhibited by other high-level languages, but must still be translated into machine language.

1. Machine Languages

In machine languages, instructions are written as sequences of 1s and 0s, called bits, that a computer can understand directly. An instruction in machine language generally tells the computer four things: (1) where to find one or two numbers or simple pieces of data in the main computer memory (Random Access Memory, or RAM), (2) a simple operation to perform, such as adding the two numbers together, (3) where in the main memory to put the result of this simple operation, and (4) where to find the next instruction to perform. While all executable programs are eventually read by the computer in machine language, they are not all programmed in machine language. It is extremely difficult to program directly in machine language because the instructions are sequences of 1s and 0s. A typical instruction in a machine language might read *10010 1100 1011* and mean add the contents of storage register A to the contents of storage register B.

executable

/ˈeksɪkjʊ:təbl/

a. 可执行的

register /ˈredʒɪstə/

n. 寄存器

storage register

存储寄存器

statement

/ˈsteɪtmənt/

n. 语句

compiler

/kəmˈpaɪlə/

n. 编译程序,

编译器

2. High-Level Languages

High-level languages are relatively sophisticated sets of statements utilizing words and syntax from human language. They are more similar to normal human languages than assembly or machine languages and are therefore easier to use for writing complicated programs. These programming languages allow larger and more complicated programs to be developed faster. However, high-level languages must be translated into machine language by another program called a **compiler** before a computer can understand them. For this reason, programs written in a high-level language may take longer to execute and use up more memory than programs written in an assembly language.

3. Assembly Languages

Computer programmers use assembly languages to make machine-language programs easier to write. In an assembly language, each statement corresponds roughly to one machine language instruction. An assembly language statement is composed with the aid of easy to remember commands. The command to add the contents of storage register A to the contents of storage register B might be written *ADD B, A* in a typical assembly language statement. Assembly languages share certain features with machine languages. For instance, it is possible to manipulate specific bits in

minimize

/'minimaɪz /

v. 使减少到
最低限度

both assembly and machine languages. Programmers use assembly languages when it is important to **minimize** the time it takes to run a program, because the translation from assembly language to machine language is relatively simple. Assembly languages are also used when some part of the computer has to be controlled directly, such as individual dots on a monitor or the flow of individual characters to a printer.

III. Classification of High-Level Languages

orient /'ɔ:riənt /v. 给...定向; 使适
应; 使面向**logic language**

逻辑语言

program module

程序模块

High-level languages are commonly classified as **procedure-oriented**, **functional**, **object-oriented**, or **logic languages**. The most common high-level languages today are procedure-oriented languages. In these languages, one or more related blocks of statements that perform some complete function are grouped together into a **program module**, or **procedure**, and given a name such as "procedure A." If the same sequence of operations is needed elsewhere in the program, a simple statement can be used to refer back to the procedure. In **essence**,¹ a procedure is just a mini-program. A large program can be constructed by grouping together procedures that perform different tasks. **Procedural languages** allow programs to be shorter and easier for the computer to read, but they require the programmer to design each procedure to be general enough to be used in different situations.

essence /'esəns /

n. 本质, 实质

procedural

language

过程语言

functional

language

函数式语言

rigorous /'rɪɡərəs /

a. 严密的; 严格的

Functional languages treat procedures like mathematical functions and allow them to be processed like any other data in a program. This allows a much higher and more **rigorous** level of program construction. Functional languages also allow variables—symbols for data that can be specified and changed by the user as the program is running—to be given values only once. This simplifies programming by reducing the need to be concerned with the exact order of statement execution, since a variable does not have to be redeclared, or restated, each time it is used in a program statement. Many of the ideas from functional languages have become key parts of many modern procedural languages.

object-oriented

language

面向对象语言

outgrowth

/'autgrəʊθ /

n. 发展结果; 产物

Object-oriented languages are **outgrowths** of functional languages. In object-oriented languages, the code used to write the program and the data processed by the program are grouped together into units called objects. Objects are further grouped into classes, which define the attributes objects must have. A simple example of a class is the class Book. Objects within

¹ in essence: 本质上, 实质上。

hierarchy

/'haɪə.rɑ:kɪ/

n. 层次, 分层 (结构), 分级 (结构)

this class might be Novel and Short Story. Objects also have certain functions associated with them, called methods. The computer accesses an object through the use of one of the object's methods. The method performs some action to the data in the object and returns this value to the computer. Classes of objects can also be further grouped into **hierarchies**, in which objects of one class can inherit methods from another class. The structure provided in object-oriented languages makes them very useful for complicated programming tasks.

deduce /dɪ'dʒu:s/

v. 推论, 推断

Logic languages use logic as their mathematical base. A logic program consists of sets of facts and if-then rules, which specify how one set of facts may be **deduced** from others, for example:

If the statement X is true, then the statement Y is false.

In the execution of such a program, an input statement can be logically deduced from other statements in the program. Many artificial intelligence programs are written in such languages.

IV. Language Structure and Components

macro /'mækrou/

n. 宏, 宏指令

Programming languages use specific types of statements, or instructions, to provide functional structure to the program. A statement in a program is a basic sentence that expresses a simple idea—its purpose is to give the computer a basic instruction. Statements define the types of data allowed, how data are to be manipulated, and the ways that procedures and functions work. Programmers use statements to manipulate common components of programming languages, such as variables and **macros** (mini-programs within a program).

data declaration

数据声明

Statements known as **data declarations** give names and properties to elements of a program called variables. Variables can be assigned different values within the program. The properties variables can have are called types, and they include such things as what possible values might be saved in the variables, how much numerical accuracy is to be used in the values, and how one variable may represent a collection of simpler values in an organized fashion, such as a table or **array**. In many programming languages, a key data type is a pointer. Variables that are pointers do not themselves have values; instead, they have information that the computer can use to locate some other variable—that is, they point to another variable.

array /ə'rei/

n. 数组; 一系列

expression

/ik'spreʃən/

n. 表达式

assignment**statement**

赋值语句

conditional**statement**

条件语句

function statement

函数语句

parameter

/pe'ræmitə/

n. 参数, 参量

segment /'segment/

n. 段; 片段;

程序段

An **expression** is a piece of a statement that describes a series of computations to be performed on some of the program's variables, such as $X+Y/Z$, in which the variables are X , Y , and Z and the computations are addition and division. An **assignment statement** assigns a variable a value derived from some expression, while **conditional statements** specify expressions to be tested and then used to select which other statements should be executed next.

Procedure and function statements define certain blocks of code as procedures or functions that can then be returned to later in the program. These statements also define the kinds of variables and parameters the programmer can choose and the type of value that the code will return when an expression accesses the procedure or function.¹ Many programming languages also permit minitranslation programs called macros. Macros translate segments of code that have been written in a language structure defined by the programmer into statements that the programming language understands.

V. History

Programming languages date back almost to the invention of the digital computer in the 1940s. The first assembly languages emerged in the late 1950s with the introduction of commercial computers. The first procedural languages were developed in the late 1950s to early 1960s: FORTRAN, created by John Backus², and then COBOL, created by Grace Hopper³. The first functional language was LISP⁴, written by John McCarthy⁵ in the late 1950s. Although heavily updated, all three languages are still widely used today.

In the late 1960s, the first object-oriented languages, such as SIMULA⁶, emerged. Logic languages became well known in the mid 1970s with the introduction of PROLOG⁷, a language used to program artificial intelligence software. During the 1970s, procedural languages continued to

¹ These statements also define the kinds of variables and parameters the programmer can choose and the type of value that the code will return when an expression accesses the procedure or function.: 这些语句也定义程序员可选的变量和参数种类, 以及表达式访问过程或函数时, 代码所返回的值的类型。

² John Backus: 约翰·巴克斯(1924—2007), 美国计算机科学家。

³ Grace Murray Hopper: 格雷斯·默里·霍珀(1906—1992), 美国海军少将, 应用数学家, 计算机程序语言的开拓者, 1991年被老布什总统授予国家科技勋章。

⁴ LISP: LISP语言, 表处理语言(List Processing的缩合)。

⁵ John McCarthy: 约翰·麦卡锡(1927—2011), 美国计算机科学家。

⁶ SIMULA: SIMULA语言, 仿真语言, 模拟语言(Simulation Language的缩合)。

⁷ PROLOG: PROLOG语言, 逻辑程序设计语言(Programming in Logic的缩合)。

merge /mə:dʒ/

v. 合并; 结合

relational language

关系语言,
相关语言

develop with ALGOL¹, BASIC, PASCAL, C, and Ada. SMALLTALK² was a highly influential object-oriented language that led to the **merging** of object-oriented and procedural languages in C++ and more recently in JAVA³. Although pure logic languages have declined in popularity, variations have become vitally important in the form of **relational languages** for modern databases, such as SQL⁴.

Exercises

I. Fill in the blanks with the information given in the text:

1. A programming language is any _____ language that can be used to write a sequence of _____ that can ultimately be processed and executed by a computer.
2. We can classify programming languages under two types: _____ languages and _____ languages.
3. A(n) _____ language is a low-level language in binary code that a computer can understand and execute directly.
4. High-level languages must first be translated into a(n) _____ language before they can be understood and processed by a computer.
5. High-level languages are commonly classified as procedure-oriented, _____, object-oriented, or _____ languages.
6. In an assembly language, each _____ corresponds roughly to one machine language instruction.
7. In procedure-oriented languages, one or more related blocks of statements that perform some complete function are grouped together into a program _____, or procedure.
8. The history of programming languages can be traced back almost to the invention of the _____ computer in the 1940s.

¹ **ALGOL**: ALGOL 语言, 算法语言 (Algorithmic Language 的缩写), 读作 /'ælgɒl/。

² **SMALLTALK**: SMALLTALK 语言, 由 IBM (国际商用机器公司) 开发, 被公认为历史上第二个面向对象的程序设计语言和第一个真正的集成开发环境。

³ **JAVA**: Java 语言, 一种新型的计算机语言, 可以跨平台运行。

⁴ **SQL**: 结构化查询语言 (Structured Query Language 的首字母缩写)。

II. Translate the following terms or phrases from English into Chinese and vice versa:

- | | |
|-------------------------------|-------------|
| 1. storage register | 11. 可执行程序 |
| 2. function statement | 12. 程序模块 |
| 3. program statement | 13. 条件语句 |
| 4. object-oriented language | 14. 赋值语句 |
| 5. assembly language | 15. 逻辑语言 |
| 6. intermediate language | 16. 机器语言 |
| 7. relational language | 17. 函数式语言 |
| 8. artificial language | 18. 程序设计语言 |
| 9. data declaration | 19. 运行计算机程序 |
| 10. Structured Query Language | 20. 计算机程序员 |

III. Fill in each of the blanks with one of the words given in the following list, making changes if necessary:

- | | | | |
|--------------------|-------------------|---------------------|------------------|
| <i>reuse</i> | <i>translate</i> | <i>step</i> | <i>memory</i> |
| <i>high-level</i> | <i>computer</i> | <i>machine</i> | <i>execution</i> |
| <i>program</i> | <i>processing</i> | <i>inconvenient</i> | <i>separate</i> |
| <i>programming</i> | <i>combine</i> | <i>programmer</i> | <i>powerful</i> |

A programming language is a language used to write instructions for the computer. It lets the programmer express data _____ in a symbolic manner without regard to machine-specific details.

The difficulty of writing programs in the _____ language of 0s and 1s led first to the development of assembly language, which allows _____ to use *mnemonics* (助记符) for instructions and symbols for variables. Such programs are then _____ by a program known as an *assembler* (汇编程序, 汇编器) into the binary encoding used by the _____. Other pieces of system software known as *linking loaders* (连接装入程序) _____ pieces of assembled code and load them into the machine's main _____ unit, where they are then ready for execution. The concept of linking _____ pieces of code was important, since it allowed "libraries" of _____ to be built up to carry out common tasks—a first _____ toward the increasingly emphasized notion of software _____. Assembly language was found to be sufficiently _____ that higher-level languages (closer to natural languages) were invented in the 1950s for easier, faster _____; along with them came the

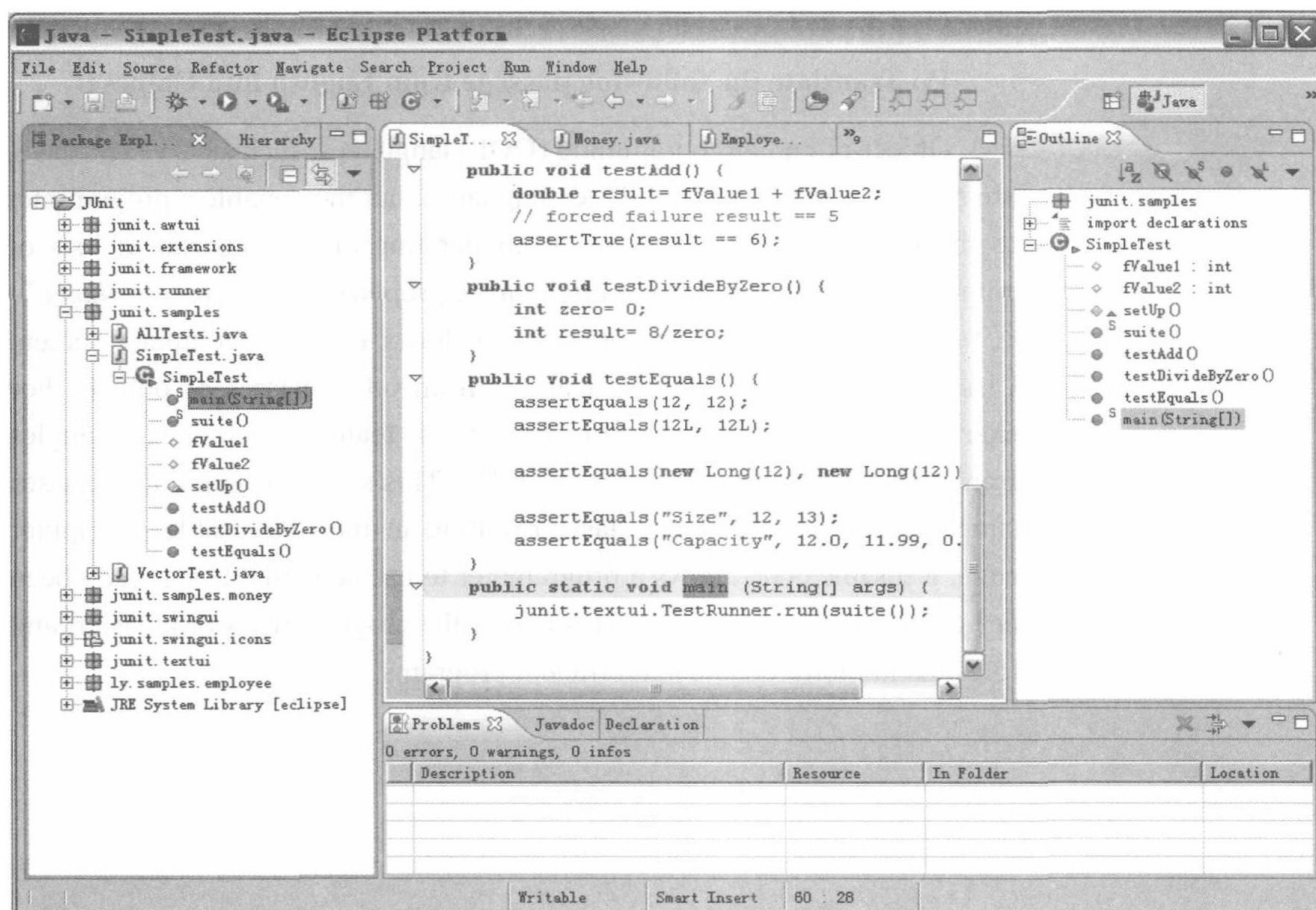
need for compilers, programs that translate _____ language programs into machine code. As programming languages became more _____ and abstract, building efficient compilers that create high-quality code in terms of _____ speed and storage consumption became an interesting computer science problem in itself.

IV. Translate the following passage from English into Chinese:

Object-oriented programming (OOP) languages, such as C++ and Java, are based on traditional high-level languages, but they enable a programmer to think in terms of collections of cooperating objects instead of lists of commands. Objects, such as a circle, have properties such as the *radius* (半径) of the circle and the command that draws it on the computer screen. Classes of objects can inherit features from other classes of objects. For example, a class defining squares can inherit features such as right angles from a class defining *rectangles* (长方形). This set of programming classes simplifies the programmer's task, resulting in more "reusable" computer code. Reusable code allows a programmer to use code that has already been designed, written, and tested. This makes the programmer's task easier, and it results in more reliable and efficient programs.

Section B

The Java Language



The Java language is a remarkable example of programming language evolution. Java builds on the familiar and useful features of C++ while removing its complex, dangerous, and superfluous elements. The result is a language that is safer, simpler, and easier to use. The following sections describe Java in contrast to C++.

superfluous

/sju:'pe:fluəs/

a. 多余的; 过剩的

instantaneous

/,instən'teinies/

a. 瞬间的, 即刻的

overhead

/'əuvəhed/

n. 经常(或管理、间接)费用

clutter /'klʌtə/

n. 凌乱, 杂乱; 杂乱的东西

1. Java Is Familiar and Simple

If you have ever programmed in C++, you will find Java's appeal to be **instantaneous**. Since Java's syntax mirrors that of C++, you will be able to write Java programs within minutes. Your first programs will come quickly and easily, with very little programming overhead.

You will have the feeling that you have eliminated a lot of **clutter** from

cryptic /'kriptik/
a. 隐秘的; 令人困惑的
header /'hedə/
n. 标题, 报头, 头标; 页眉
preprocessor
/pri:'preusesə/
n. 预处理程序, 预处理器
arcane /ɑ:'kein/
a. 神秘的, 晦涩难解的
delve /delv/
v. 搜索, 翻查
package /'pækidʒ/
n. 程序包, 软件包
software package
软件包
interpreter
/in'te:prite/
n. 解释程序, 解释器

union /'ju:njən/
n. 共用体(定义关键字), 共用(数据类型)

class hierarchy
类层次
inheritance

/in'heritəns/
n. 继承(性)
encapsulation
/in,kæpsju'leifən; -sə-/
n. 封装

polymorphism
/pɒlɪ'mɔ:fizəm/
n. 多态性, 多形性
clarity /'klærɪti/
n. 清晰, 明晰

your programs—and you will have. All the **cryptic** header files and **preprocessor** statements of C and C++ are gone. All the **arcane** **#define**¹ statements and **typedefs**² have been taken away. You will no longer have to **delve** through several levels of header files to correctly reference **API**³ calls. And no one will have to suffer to figure out how to use your software.

Java programs simply import the **software packages** they need. These packages may be in another directory, on another drive, or on a machine on the other side of the Internet. The Java compiler and **interpreter** figure out what objects are referenced and supply the necessary linkage.

2. Java Is Object-Oriented

If you think C++ is an object-oriented programming language, you are in for a big surprise⁴. After using Java to write a few programs, you'll get a better feeling for what object-oriented software is all about.

Java deals with classes and objects, pure and simple⁵. They aren't just more data structures that are available to the programmer—they are the basis for the entire programming language.

In C++, you can declare a class, but you don't have to. You can declare a structure or a **union** instead. You can declare a whole bunch of loosely associated variables and use them with C-style functions. In Java, classes and objects are at the center of the language. Everything else revolves around them. You can't declare functions and procedures. They don't exist. You can't use structures, unions, or **typedefs**. They're gone, too. You either use classes and objects or you don't use Java. It's that simple.

Java provides all the luxuries of object-oriented programming: **class hierarchy**, **inheritance**, **encapsulation**, and **polymorphism**—in a context that is truly useful and efficient.

The main reason for developing object-oriented software, besides **clarity** and simplicity, is the desperate hope that somehow the objects you develop will be reused. Java not only encourages software reuse, it demands it. To write any sort of Java program, no matter how simple, you must build

¹ **#define**: C 语言中的特定字符, 只有编程上的含义。

² **typedef**: C 语言中的特定字符, 只有编程上的含义。

³ **API**: 应用程序接口 (Application Program Interface 的首字母缩略)。

⁴ *you are in for a big surprise*: 你一定会大吃一惊。句中的 be in for 为固定搭配, 意为“肯定会经历”、“注定要遭受”。

⁵ *pure and simple*: 纯粹的, 不折不扣的。

on the classes and methods of the Java API.

Once you have begun developing software in Java, you have two choices:

- Build on the classes you have developed, thereby reusing them.
- Rewrite your software from scratch, copying and tailoring useful parts of existing software.

appealing /ə'pi:liŋ/
a. 吸引人的; 有感染力的

With Java, the temptation to start from scratch is no longer **appealing**. Java's object-oriented structure forces you to develop more useful, more tailorable, and much simpler software the first time around¹.

3. Java Is Safer and More Reliable

Java is safer to use than C++ because it keeps you from doing the things that you do badly, while making it easier to do the things that you do well.

assembly code
汇编代码

Java won't automatically convert data types. You have to explicitly convert from one class to another. C++, under the most undesirable conditions, will automatically convert one type to another. It has all the flexibility of **assembly code**. Java doesn't assume that you know what you are doing. It makes sure that you do.

C++ pointers don't exist in Java. You can no longer access objects indirectly or by chance. You don't need to. You declare objects and reference those objects directly. Complex pointer arithmetic is avoided. If you need an indexed set of objects, you can use an array of objects. The concept of "the address of an object" is eliminated from the programming model, and another assembly language dinosaur is laid to rest. As a result, it becomes much easier to do things correctly in Java.

runtime /'rʌntaim/
n. 运行时刻
syntactic
/sin'tæktik/
a. (按照) 句法的
semantically
/si'mæntikəli/
ad. 在语义上
mom /mɒm/
n. (主美口) 妈妈

Java's reliability extends beyond the language level to the compiler and the runtime system. Compile-time checks identify many programming errors that go undetected in other programming languages. These checks go beyond syntactic checking to ensure that statements are **semantically correct**.

Runtime checks are also more extensive and effective. Remember your teacher or mom telling you to "Check your work twice to make sure it's

¹ *the first time around*: 这里的 around 为副词, 用于描述 something that has happened before or things that happen regularly, 只是表示所描述的事情以前发生过或经常发生, 省略了对意思影响不大。

linker /'liŋkə/
n. 连接程序,
链接程序
bounds checking
边界检查

flaw /flo:/
n. 缺点, 瑕疵
spoof /spu:f/
v. 哄骗, 欺骗
browse /brauz/
v. 浏览
unauthorized
/ˌʌn'ɔ:θəraɪzd/
a. 未经授权的
vulnerable
/'vʌlnərəbl/
a. 脆弱的; 易受攻击的
constrain
/kən'streɪn/
v. 约束, 限制
applet /'æplət/
n. 小应用程序
Trojan /'trəʊdʒən/
(horse)
n. 特洛伊木马程序
(或病毒)
malicious /mə'liʃəs/
a. 恶意的
bytecode /'baɪtkəʊd/
n. 字节码
theorem /'θiərəm/
n. 定理
prover /'pru:və/
n. 证明程序
theorem prover
定理证明程序,
定理证明器

segregate /'segrɪgeɪt/
v. 隔离; 分开
trustworthy
/'trʌst.wə:ði/
a. 值得信任的;
可信的; 可靠的

right”? The Java linker understands class types and performs compiler-level type checking, adding redundancy to reliability. It also performs bounds checking and eliminates indirect object access, even under error conditions.

4. Java Is Secure

If you gave a skilled hacker a program written in C or C++ and told him to find any security flaws, there are half a dozen things that he would immediately look for: gaining access to the operating system, causing an unexpected return of control, overwriting critical memory areas, acquiring the ability to spoof or modify other programs, browsing for security information, and gaining unauthorized access to the file system.

Why is C or C++ more vulnerable than Java? When a programmer develops software, he or she usually focuses on how to get the software to work correctly and efficiently. C and C++ do not constrain the programmer from meeting these goals and provide a number of flexible features that enable the programmer to meet his end. The hacker is also able to take advantage of these features and use them in ways that weren't originally intended, causing the undesirable consequences identified in the previous paragraph. In short, C and C++ provide a great offense, but no defense. Java, on the other hand, is defensive by nature. Every time a Java-enabled browser downloads a compiled Java class, such as an applet, it runs the risk of running Trojan horse code. Because of this ever-present threat, it subjects the code to a series of checks that ensure that it is correct and secure.

The Java runtime system is designed to enforce a security policy that prevents execution of malicious code. It does this by remembering how objects are stored in memory and enforcing correct and secure access to those objects according to its security rules. It performs bytecode verification by passing compiled classes through a simple theorem prover that either proves that the code is secure or prevents the code from being loaded and executed. The class is Java's basic execution unit and security is implemented at the class level.

The Java runtime system also segregates software according to its origin. Classes from the local system are processed separately from those of other systems. This prevents remote systems from replacing local system software with code that is less trustworthy.

dialog box

对话框

multithreaded

/'mʌltiθredɪd/

a. 多线程的

multithreading

/'mʌltiθredɪŋ/

n. 多线程操作

thread /θred/

n. 线程, 线索

duke /dʒʊ:k/

n. 公爵

convoluted

/'kɒnvəlu:tɪd/

a. 盘绕的; 盘错的, 错综复杂的

gaggle /'gægl/

n. (紊乱而有联系的) 一堆

intertwine

/,ɪntə'twɛɪn/

v. (使) 缠结, (使) 缠绕在一起

synchronization

/,sɪŋkrənaɪ'zeɪʃən/

n. 同步(化)

compiled code

编译执行的代码

interpreted code

解释执行的代码

field /fi:ld/

v. 派...上场; 实施

native code

本机(代)码

underway

/'ʌnde'wei/

a. 在进行中的

Java-enabled browsers, such as HotJava¹, allow the user to control the accesses that Java software may make of the local system. When a Java applet needs permission to access local resources, such as files, a security **dialog box** is presented to the user, requesting explicit user permission. This "Mother may I?" approach ensures that the user always has the final say in the security of his system.

5. Java Is Multithreaded

Java, like Ada, and unlike other languages, provides built-in language support for **multithreading**. Multithreading allows more than one thread of execution to take place within a single program. This allows your program to do many things at once: make the Duke dance, play his favorite tune, and interact with the user, seemingly all at the same time. Multithreading is an important asset because it allows the programmer to write programs as independent threads, rather than as a **convoluted gaggle** of intertwined activities. Multithreading also allows Java to use idle CPU time to perform necessary garbage collection and general system maintenance, enabling these functions to be performed with less impact on program performance.

Writing multithreaded programs is like dating several people concurrently. Everything works fine until the threads start to interact with each other in unexpected ways. Java provides the support necessary to make multithreading work safely and correctly. Java supports multithreading by providing **synchronization** capabilities that ensure that threads share information and execution time in a way that is thread safe.

6. Java Is Interpreted and Portable

While it is true that compiled code will almost always run more quickly than interpreted code, it is also true that interpreted code can usually be developed and fielded more inexpensively, more quickly, and in a more flexible manner. It is also usually much more portable.

Java, in order to be a truly platform-independent programming language, must be interpreted. It does not run as fast as compiled native code, but it doesn't run much slower, either. For the cases where execution in native machine code is absolutely essential, work is underway to translate Java bytecode into machine code as it is loaded.

¹ HotJava: 由美国 Sun 公司(太阳微系统公司)开发的支持 Java 开发环境的 Internet 浏览器。

outweigh /,aut'wei/
v. 在价值(或重要性、影响等)方面超过

source code
源(代)码

The advantages of being interpreted outweigh any performance impacts. Because Java is interpreted, it is much more portable. If an operating system can run the Java interpreter and support the Java API, then it can faithfully run all Java programs.

Interpreted programs are much more easily kept up-to-date. You don't have to recompile them for every change. In Java, recompilation is automatic. The interpreter detects the fact that a program's bytecode file is out-of-date with respect to its source code file and recompiles it as it is loaded.

Because of Java's interpreted nature, linking is also more powerful and flexible. Java's runtime system supports dynamic linking between local class files and those that are downloaded from across the Internet. This feature provides the basis for Web programming.

7. Java Is the Programming Language of the Web

de facto /di:'fæktəu/
〈拉〉实际的, 事实上的
offshoot /'ɒfʃu:t/
n. 支族, 旁系; 衍生事物
release /ri:'li:s/
n. (程序或软件的) 版本; 发布
vendor /'vendə/
n. 卖主; 厂家, 厂商
broker /'brəʊkə/
n. 代理者; 代理程序

Java has become the de facto programming language of the Web. It is being licensed by nearly every major software company. It has some offshoots and potential competition, such as JavaScript¹ and VBScript², but it remains the first Web programming language and the most powerful language for developing platform-independent software.

Java is also evolving beyond the Web and becoming a key component in distributed application development. Some releases of Sun's products emphasize Java's importance to distributed object-based software development. Several other vendors have introduced products that enable Java to be integrated into the Common Object Request Broker Architecture (CORBA³), which is the framework for distributed object communication.

Exercises

I. Fill in the blanks with the information given in the text:

1. Java is an object-oriented programming language, for which classes and _____ are the basis.

¹ JavaScript: Java 脚本语言。

² VBScript: Visual Basic 脚本语言。

³ CORBA: 公用对象请求代理(程序)体系结构(Common Object Request Broker Architecture 的首字母缩略)。

2. Java is designed to be _____, which makes it a useful language for programming Web applications, since users access the Web from many types of computers.

3. Java provides built-in language support for _____. That is, it allows more than one thread of execution to take place within a single program.

4. The Java _____ system is designed to enforce a security policy that prevents execution of malicious code.

II. Translate the following terms or phrases from English into Chinese and vice versa:

- | | |
|--------------------------|------------|
| 1. native code | 6. 机器码 |
| 2. header file | 7. 汇编码 |
| 3. multithreaded program | 8. 特洛伊木马程序 |
| 4. Java-enabled browser | 9. 软件包 |
| 5. malicious code | 10. 类层次 |

Section C

Arrays

Imagine you have a problem that requires 20 numbers to be processed. You need to read them, process them, and print them. You must also keep these 20 numbers in memory for the duration of the program. You can define 20 variables, each with a different name, as shown in Figure 3C-1.

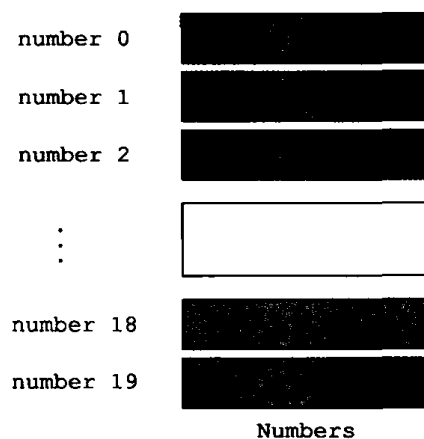


Figure 3C-1: Twenty Individual Variables

But having 20 different names creates another problem. How can you read 20 numbers from the keyboard and store them? To read 20 numbers from the keyboard, you need 20 references, one to each variable.¹ Furthermore, once you have them in memory, how can you print them? To print them, you need another 20 references. In other words, you need the **flowchart** in Figure 3C-2 to read, process, and print these 20 numbers.

Although this may be acceptable for 20 numbers, it is definitely not acceptable for 200 or 2000 or 20,000 numbers. To process large amounts of data, you need a powerful data structure such as an array. An array is a fixed-size, sequenced collection of elements of the same data type. Since an array is a sequenced collection, you can refer to the elements in the array as the first element, the second element, and so forth until you get to the last element. If you were to put your 20 numbers into an array, you could

reference

/ˈrefərəns/

n. 引用, 参考;
基准

flowchart

/ˈfləʊtʃɑ:t/

n. 流程图

¹ To read 20 numbers from the keyboard, you need 20 references, one to each variable.: 为了从键盘上读入20个数, 你需要20个引用, 每个引用对应着一个变量。

designate

/'deizineit/

v. 指定, 指明;

命名; 指派

subscript

/'sʌbskript/

n. 下标, 角注

ordinal /'ɔ:dinal/

a. 顺序的

ordinal number

序数

designate the first element number_0 as shown in Figure 3C-1. In a similar fashion, you could refer to the second number as number_1 and the third number as number_2 . Continuing the series, the last number would be number_{19} . The **subscripts** indicate the **ordinal number** of the element counting from the beginning of the array.¹

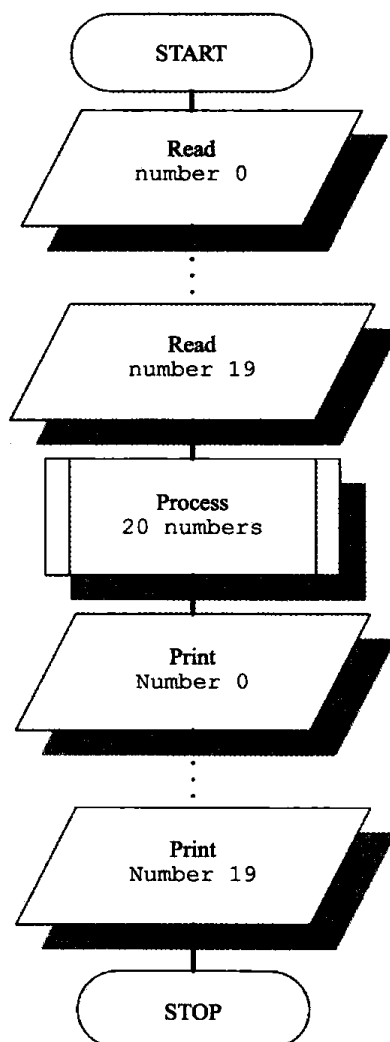


Figure 3C-2: Processing Individual Variables

What you have seen is that the elements of the array are individually **addressed** through their subscripts (Figure 3C-3).² The array as a whole has a name, number , but each member can be accessed individually using its subscript.

The advantages of the array would be limited if you didn't also have

address /ə'dres/

v. 编址; 寻址

¹ *The subscripts indicate the ordinal number of the element counting from the beginning of the array.*: 下标表示元素从数组开头数的序数。

² *the elements of the array are individually addressed through their subscripts*: 数组的元素通过其下标给出各自的地址。

**programming
construct**
编程结构

programming constructs that allow you to process the data more conveniently. Fortunately, there is a powerful set of programming constructs, loops, that makes array processing easy.

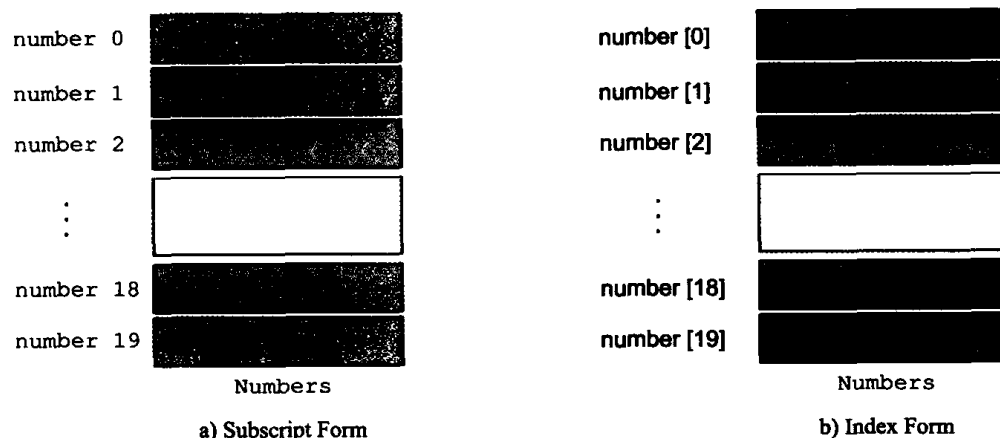


Figure 3C-3: Arrays with Subscripts and Indexes

You can use loops to read and write the elements in an array. You can use loops to add, subtract, multiply, and divide the elements. You can also use loops for more complex processing such as calculating averages. Now it does not matter if there are 2, 20, 200, 2000, or 20,000 elements to be processed. Loops make it easy to handle them all.

But one question still remains: How can you write an instruction so that one time it refers to the first element of an array and the next time it refers to another element? It is really quite simple: You simply borrow from the subscript concept you have been using. Rather than using subscripts, however, you place the subscript value in square brackets. Using this notation, you refer to `number0` as `number[0]`.

Following the convention, `number1` becomes `number[1]` and `number19` becomes `number[19]`. This is known as **indexing**. Using a typical reference, you now refer to your array using the variable.¹ The flowchart to process your 20 numbers using an array and looping is in Figure 3C-4.

In the following section, we study one array application: the **frequency array** and its graphical representation.

bracket /'brækit/
n. 括号

square bracket
方括号

notation

/nəu'teɪʃən/

n. 标记法; 记号

indexing /'indeksɪŋ/
n. 编索引; 标引;
加下标; 变址

looping /'lu:pɪŋ/

n. 循环; 构成环形

frequency array
频率数组

¹ Using a typical reference, you now refer to your array using the variable.: 使用一个典型的引用, 你就可以使用变量来访问数组。

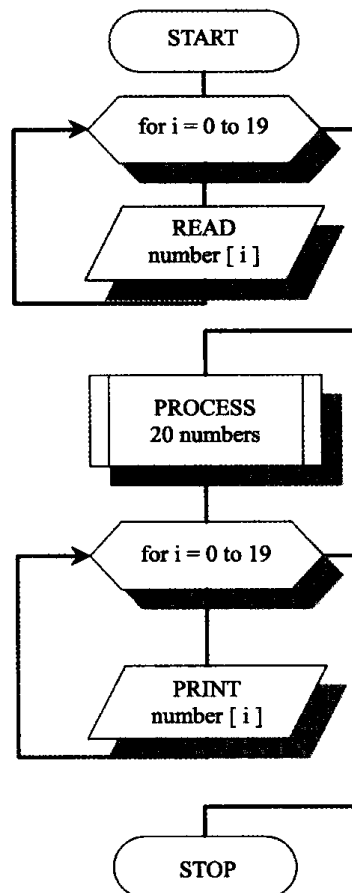


Figure 3C-4: Processing an Array

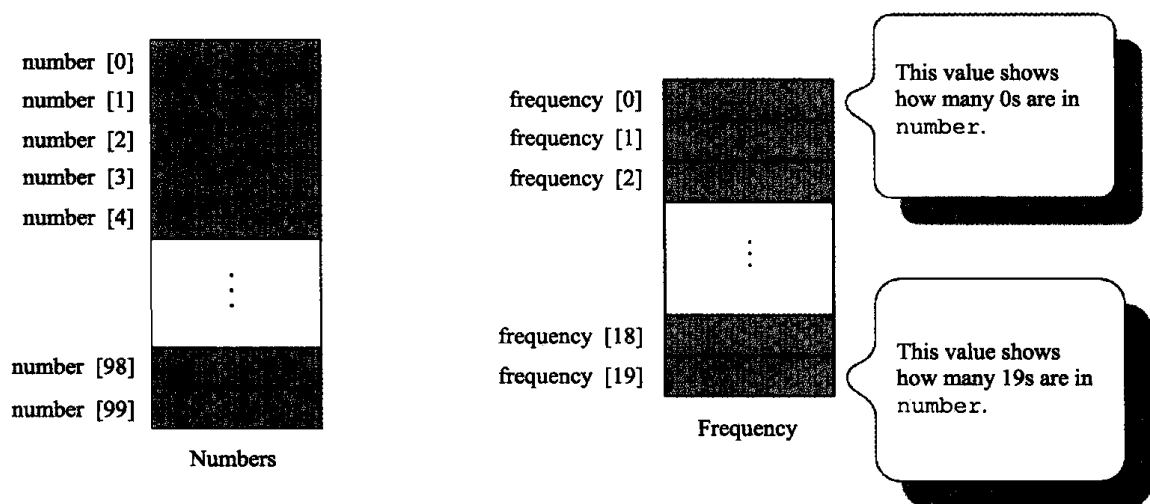


Figure 3C-5: Frequency Array

A frequency array shows the number of elements with the same value found in a series of numbers. For example, suppose you have taken a sample of 100 values between 0 and 19. You want to know how many of the values are 0, how many are 1, how many are 2, and so forth up through 19.

histogram

/'hɪstəʊgræm/

n. 直方图, 矩形图;
频率分布图**bar** /bɑː/

n. 条; 条形图

bar chart

条形图

asterisk /'æstərɪsk/

n. 星号

You can read these values into an array called `numbers`. Then you create an array of 20 elements that will show the frequency of each value in the series (Figure 3C-5).

A **histogram** is a pictorial representation of a frequency array. Instead of printing the values of the elements to show the frequency of each number, you print a histogram in the form of a **bar chart**. For example, Figure 3C-6 is a histogram for a set of numbers in the range 0 to 19. In this example, **asterisks (*)** are used to build the bar. Each asterisk represents one occurrence of the data value.

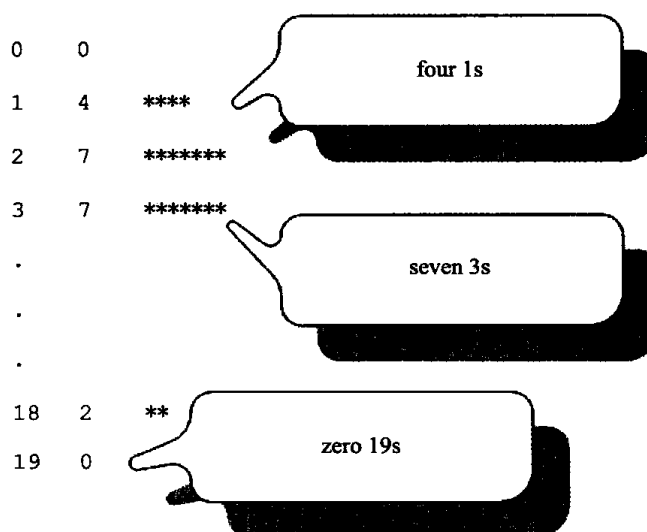


Figure 3C-6: Histogram

one-dimensional

array 一维数组

linear /'lɪniə/a. 线(性)的;
直线的**two-dimensional**

array 二维数组

multidimensional

array 多维数组

The arrays discussed so far are known as **one-dimensional arrays** because the data are organized **linearly** in only one direction. Many applications require that data be stored in more than one dimension. One common example is a table, which is an array that consists of rows and columns. Figure 3C-7 shows a table, which is commonly called a **two-dimensional array**. Note that arrays can have three, four, or more dimensions. However, the discussion of **multidimensional arrays** is beyond the scope of this section.

The indexes in the definition of a two-dimensional array represent rows and columns.¹ This format maps the way the data are laid out in memory.² If you were to consider memory as a row of bytes with the

¹ *The indexes in the definition of a two-dimensional array represent rows and columns.*: 定义二维数组的索引分别表示行和列。

² *This format maps the way the data are laid out in memory.*: 这种格式映射了数据在存储器中的存储方式。

lowest address on the left and the highest address on the right, then an array would be placed in memory with the first element to the left and the last element to the right. Similarly, if the array is two-dimensional, then the first dimension is a row of elements that is stored to the left. This is known as “row-major” storage (Figure 3C-8).¹

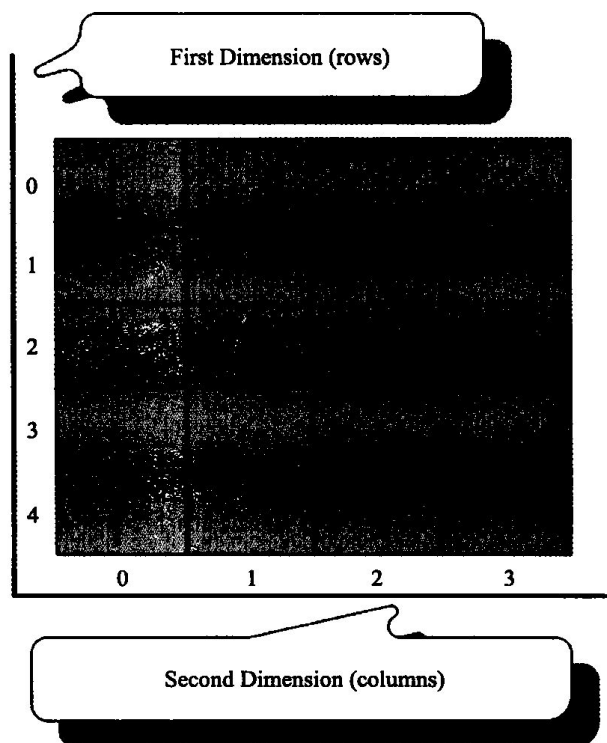
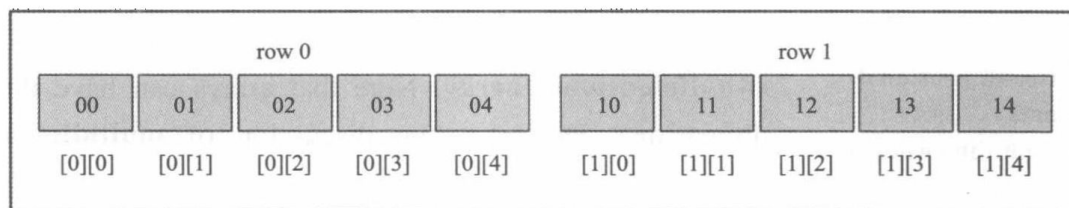


Figure 3C-7: Two-Dimensional Array

00	01	02	03	04
10	11	12	13	14

User's View



Memory View

Figure 3C-8: Memory Layout

view / vju: /
n. 视图

¹ This is known as “row-major” storage.: 这通常称为“以行为主的”存储。row-major 表示“以行为主的”、“行优先的”。

Exercises

I. Fill in the blanks with the information given in the text:

1. Given the array called `object` with 20 elements, if you see the term `object10`, you know the array is in _____ form; if you see the term `object[10]`, you know the array is in _____ form.
2. In most programming languages, an array is a static data structure. When you define an array, the size is _____.
3. A(n) _____ is a pictorial representation of a frequency array.
4. An array that consists of just rows and columns is probably a(n) _____ array.

II. Translate the following terms or phrases from English into Chinese and vice versa:

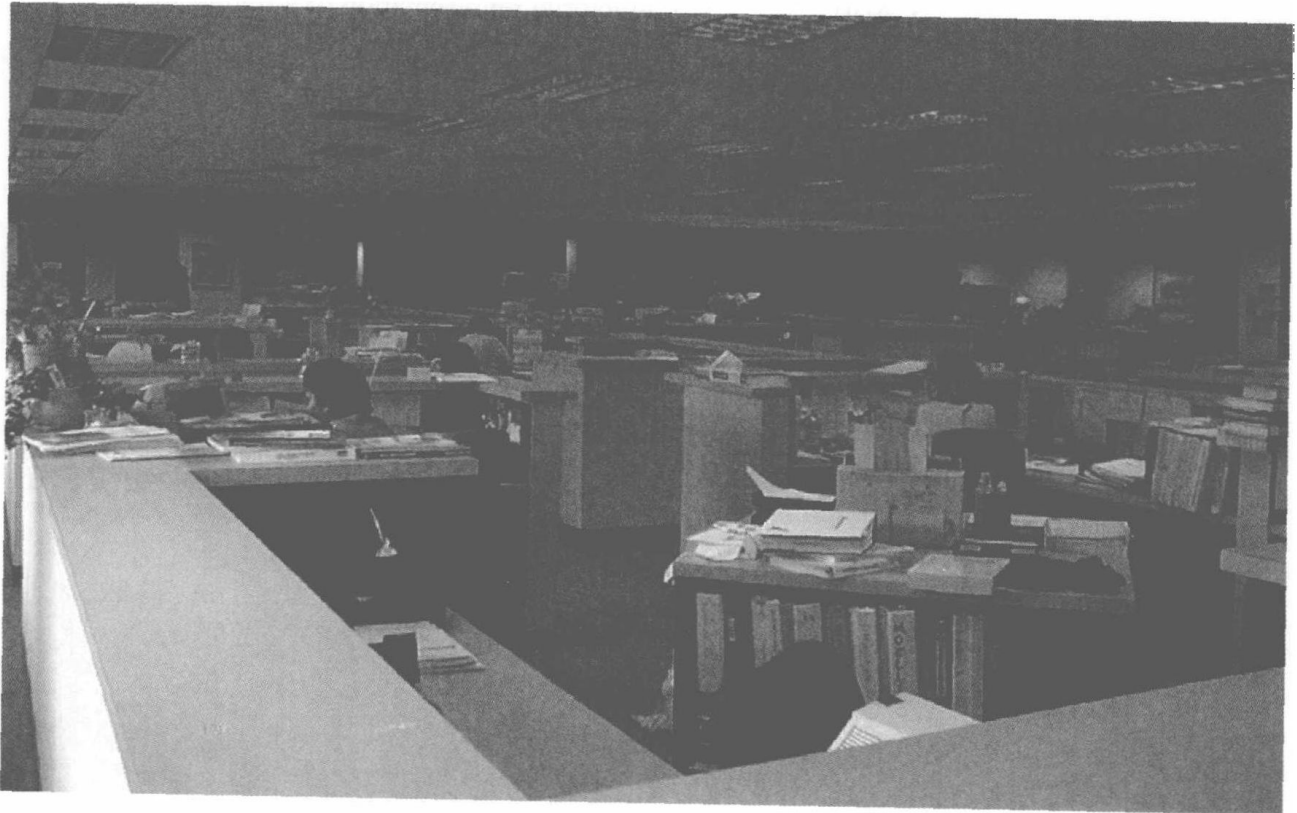
- | | |
|-----------------------------|---------|
| 1. bar chart | 5. 用户视图 |
| 2. frequency array | 6. 下标形式 |
| 3. graphical representation | 7. 一维数组 |
| 4. multidimensional array | 8. 编程结构 |

Unit 4 Software Development

(软件开发)

Section A

Computer Program



I. Introduction

A computer program is a set of instructions that directs a computer to perform some processing function or combination of functions. For the instructions to be carried out, a computer must execute a program, that is, the computer reads the program, and then follows the steps **encoded** in the program in a precise order until completion. A program can be executed many different times, with each execution yielding a potentially different result depending upon the options and data that the user gives the computer.

encode /en'kəud /
v. 把...编码; 把...
译成电码(或密码)

Programs fall into two major classes: application programs and

operating systems. An application program is one that carries out some function directly for a user, such as word processing or game playing. An operating system is a program that manages the computer and the various resources and devices connected to it, such as RAM (random access memory), hard drives, monitors, keyboards, printers, and modems, so that they may be used by other programs.

II. Program Development

Software designers create new programs by using special applications programs, often called **utility programs** or **development programs**. A programmer uses another type of program called a **text editor** to write the new program in a special notation called a **programming language**. With the text editor, the programmer creates a text file, which is an ordered list of instructions, also called the **program source file**. The individual instructions that make up the program source file are called **source code**. At this point, a special applications program translates the source code into **machine language**, or **object code**—a format that the operating system will recognize as a proper program and be able to execute.

Three types of applications programs translate from source code to object code: **compilers**, **interpreters**, and **assemblers**. The three operate differently and on different types of programming languages, but they serve the same purpose of translating from a programming language into machine language.

A compiler translates text files written in a high-level programming language—such as FORTRAN, C, or Pascal—from the source code to the object code all at once. This differs from the approach taken by **interpreted languages** such as BASIC, in which a program is translated into object code statement by statement as each instruction is executed. The advantage of interpreted languages is that they can begin executing the program immediately instead of having to wait for all of the source code to be compiled. Changes can also be made to the program fairly quickly without having to wait for it to be compiled again. The disadvantage of interpreted languages is that they are slow to execute, since the entire program must be translated one instruction at a time, each time the program is run. On the other hand, **compiled languages** are compiled only once and thus can be executed by the computer much more quickly than interpreted languages. For this reason, compiled languages are more common and are almost always used in professional and scientific applications.

utility /ju:'tiliti/

n. 实用程序,
公用程序

utility program

实用程序,
公用程序

text editor

文本编辑程序,
文本编辑器

source file

源文件

object code

目标(代)码

assembler

/ə'sembə/

n. 汇编程序,
汇编器

interpreted

language

解释执行的语言

compiled language

编译执行的语言

Another type of translator is the assembler, which is used for programs or parts of programs written in assembly language. Assembly language is another programming language, but it is much more similar to machine language than other types of high-level languages. In assembly language, a single statement can usually be translated into a single instruction of machine language. Today, assembly language is rarely used to write an entire program, but is instead most often used when the programmer needs to directly control some aspect of the computer's function.

Programs are often written as a set of smaller pieces, with each piece representing some aspect of the overall application program. After each piece has been compiled separately, a program called a linker combines all of the translated pieces into a single executable program.

Programs seldom work correctly the first time, so a program called a **debugger** is often used to help find problems called **bugs**. **Debugging** programs usually detect an event in the executing program and point the programmer back to the origin of the event in the program code.

Recent programming systems, such as Java, use a combination of approaches to create and execute programs. A compiler takes a Java **source program** and translates it into an intermediate form. Such intermediate programs are then transferred over the Internet into computers where an **interpreter program** then executes the intermediate form as an application program.

III. Program Elements

Most programs are built from just a few kinds of steps that are repeated many times in different contexts and in different combinations throughout the program. The most common step performs some computation, and then proceeds to the next step in the program, in the order specified by the programmer.

Programs often need to repeat a short series of steps many times, for instance in looking through a list of game scores and finding the highest score. Such repetitive sequences of code are called loops.

One of the capabilities that make computers so useful is their ability to make conditional decisions and perform different instructions based on the values of data being processed. *If-then-else* statements implement this function by testing some piece of data and then selecting one of two

debugger / di:'bʌgə /

n. 调试程序,
排错程序

bug / bʌg /

n. (程序) 错误,
故障

debug / di:'bʌg /

v. 调试, 排除 (程
序) 中的错误

source program

源程序

interpreter

program

解释程序

sequences of instructions on the basis of the result. One of the instructions in these alternatives may be a *goto* statement that directs the computer to select its next instruction from a different part of the program. For example, a program might compare two numbers and branch to a different part of the program depending on the result of the comparison:

If x is greater than y
 then
 goto instruction #10
 else continue

Programs often use a specific sequence of steps more than once. Such a sequence of steps can be grouped together into a **subroutine**, which can then be called, or accessed, as needed in different parts of the main program. Each time a subroutine is called, the computer remembers where it was in the program when the call was made, so that it can return there upon completion of the subroutine. **Preceding** each call, a program can specify that different data be used by the subroutine, allowing a very general piece of code to be written once and used in multiple ways.

Most programs use several varieties of subroutines. The most common of these are functions, procedures, **library routines**, **system routines**, and device drivers. Functions are short subroutines that compute some value, such as computations of angles, which the computer cannot compute with a single basic instruction. Procedures perform a more complex function, such as sorting a set of names. Library routines are subroutines that are written for use by many different programs. System routines are similar to library routines but are actually found in the operating system. They provide some service for the application programs, such as printing a line of text. Device drivers are system routines that are added to an operating system to allow the computer to communicate with a new device, such as a scanner, modem, or printer. Device drivers often have features that can be executed directly as applications programs. This allows the user to directly control the device, which is useful if, for instance, a color printer needs to be **realigned** to attain the best printing quality after changing an **ink cartridge**.

IV. Program Function

Modern computers usually store programs on some form of magnetic storage media that can be accessed randomly by the computer, such as the hard drive disk permanently located in the computer, or a portable floppy

subroutine

/ˈsʌbruːtiːn/

n. 子例程,
子(例行)程序

precede

/ˌpriːˈsiːd, pri-/

v. 处在...之前,
先于

routine /ruːˈtiːn/

n. 例程, 例行程序

library routine

库程序, 库存(子)
程序, 程序库例行
程序

system routine

系统例(行)程(序)

realign /ˌriːəˈleɪn/

v. 重新调整

cartridge /ˈkɑːtrɪdʒ/

n. 盒, 匣

ink cartridge

墨盒

disk. Additional information on such disks, called directories, indicates the names of the various programs on the disk, when they were written to the disk, and where the program begins on the disk media. When a user directs the computer to execute a particular application program, the operating system looks through these directories, locates the program, and reads a copy into RAM. The operating system then directs the CPU (central processing unit) to start executing the instructions at the beginning of the program. Instructions at the beginning of the program prepare the computer to process information by locating free **memory locations** in RAM to hold working data, retrieving copies of the standard options and **defaults** the user has indicated from a disk, and drawing initial displays on the monitor.

memory location
存储单元

default / di'fɔ:lt /
n. 默认, 缺省,
系统设定值

The application program requests a copy of any information the user enters by making a call to a system routine. The operating system converts any data so entered into a standard internal form. The application then uses this information to decide what to do next—for example, perform some desired processing function such as **reformatting** a page of text, or obtain some additional information from another file on a disk. In either case, calls to other system routines are used to actually carry out the display of the results or the accessing of the file from the disk.

reformat
/ ,ri:'fɔ:mæt /
v. 重新格式化

prompt / prɒmpt /
v. 提示, 提醒
system call
系统调用

When the application reaches completion or is **prompted** to quit, it makes further **system calls** to make sure that all data that needs to be saved has been written back to disk. It then makes a final system call to the operating system indicating that it is finished. The operating system then frees up the RAM and any devices that the application was using and awaits a command from the user to start another program.

V. History

People have been storing sequences of instructions in the form of a program for several centuries. Music boxes of the 18th century and **player pianos** of the late 19th and early 20th centuries played musical programs stored as series of metal pins, or holes in paper, with each line (of pins or holes) representing when a note was to be played, and the pin or hole indicating what note was to be played at that time. More elaborate control of physical devices became common in the early 1800s with French inventor Joseph-Marie Jacquard's invention of the **punch-card** controlled weaving loom. In the process of weaving a particular pattern, various parts of the loom had to be mechanically positioned. To automate this process, Jacquard used a single paper card to represent each positioning of the loom,

player piano
自动钢琴

punch card
穿孔卡片

tapestry /'tæpɪstri/
n. 花毯; 挂毯

with holes in the card to indicate which loom actions should be done. An entire **tapestry** could be encoded onto a deck of such cards, with the same deck yielding the same tapestry design each time it was used. Programs of over 24,000 cards were developed and used.

counterpart
/'kauntəpɑ:t/
n. 对应的物(或人)

The world's first programmable machine was designed—although never fully built—by the English mathematician and inventor, Charles Babbage. This machine, called the Analytical Engine, used punch cards similar to those used in the Jacquard loom to select the specific arithmetic operation to apply at each step. Inserting a different set of cards changed the computations the machine performed. This machine had **counterparts** for almost everything found in modern computers, although it was mechanical rather than electrical. Construction of the Analytical Engine was never completed because the technology required to build it did not exist at the time.

The first card deck programs for the Analytical Engine were developed by British mathematician Augusta Ada Byron, daughter of the poet Lord Byron¹. For this reason she is recognized as the world's first programmer.

The modern concept of an internally stored computer program was first proposed by Hungarian-American mathematician John von Neumann in 1945. Von Neumann's idea was to use the computer's memory to store the program as well as the data. In this way, programs can be viewed as data and can be processed like data by other programs. This idea greatly simplifies the role of program storage and execution in computers.

VI. The Future

The field of computer science has grown rapidly since the 1950s due to the increase in their use. Computer programs have undergone many changes during this time in response to user need and advances in technology. Newer ideas in computing such as parallel computing, distributed computing, and artificial intelligence, have radically altered the traditional concepts that once determined program form and function.

Computer scientists working in the field of parallel computing, in which multiple CPUs cooperate on the same problem at the same time, have introduced a number of new program models. In parallel computing parts of a problem are worked on simultaneously by different processors, and this speeds up the solution of the problem. Many challenges face scientists and engineers who design programs for parallel processing computers, because

¹ Lord Byron: 拜伦勋爵(1788—1824), 英国诗人, 全名 George Gordon Byron。

of the extreme complexity of the systems and the difficulty involved in making them operate as effectively as possible.

Another type of parallel computing called distributed computing uses CPUs from many interconnected computers to solve problems. Often the computers used to process information in a distributed computing application are connected over the Internet. Internet applications are becoming a particularly useful form of distributed computing, especially with programming languages such as Java. In such applications, a user logs on¹ to a Web site and downloads a Java program onto their computer. When the Java program is run, it communicates with other programs at its home Web site, and may also communicate with other programs running on different computers or Web sites.

inference engine
推理机

Research into artificial intelligence (AI) has led to several other new styles of programming. Logic programs, for example, do not consist of individual instructions for the computer to follow blindly, but instead consist of sets of rules: if x happens then do y. A special program called an **inference engine** uses these rules to “reason” its way to a conclusion when presented with a new problem. Applications of logic programs include automatic monitoring of complex systems, and proving mathematical theorems.

A radically different approach to computing in which there is no program in the conventional sense is called a neural network. A neural network is a group of highly interconnected simple processing elements, designed to mimic the brain. Instead of having a program direct the information processing in the way that a traditional computer does, a neural network processes information depending upon the way that its processing elements are connected. Programming a neural network is accomplished by presenting it with known patterns of input and output data and adjusting the relative importance of the interconnections between the processing elements until the desired pattern matching is accomplished. Neural networks are usually simulated on traditional computers, but unlike traditional computer programs, neural networks are able to learn from their experience.

¹ *log on*: 登录, 注册, 进入系统。

Exercises

I. Fill in the blanks with the information given in the text:

1. Computer programs fall into two major classes: _____ programs and _____ systems.
2. There are three types of application programs to translate source code into object code. They are compilers, interpreters, and _____.
3. A(n) _____ translates all the source code of a program written in a high-level language into object code prior to the execution of the program.
4. In the case of a(n) _____, a program is translated into executable form and executed one statement at a time rather than being translated completely before execution.
5. A(n) _____ is a program that is often used to help find problems in other programs.
6. A(n) _____ is a sequence of code in a program executed repeatedly, either for a fixed number of times or until a certain condition is met.
7. When you install a new device in a computer, you have to add the correct device _____ to the operating system to allow the computer to communicate with the device.
8. The modern concept of an internally stored computer program was first proposed by _____ in 1945.

II. Translate the following terms or phrases from English into Chinese and vice versa:

- | | |
|------------------------|-------------|
| 1. inference engine | 11. 解释执行的语言 |
| 2. system call | 12. 设备驱动程序 |
| 3. compiled language | 13. 源程序 |
| 4. parallel computing | 14. 调试程序 |
| 5. pattern matching | 15. 目标代码 |
| 6. memory location | 16. 应用程序 |
| 7. interpreter program | 17. 实用程序 |
| 8. library routine | 18. 逻辑程序 |

9. intermediate program

19. 墨盒

10. source file

20. 程序的存储与执行

III. Fill in each of the blanks with one of the words given in the following list, making changes if necessary:

<i>similar</i>	<i>computer</i>	<i>directly</i>	<i>compiler</i>
<i>produce</i>	<i>engineer</i>	<i>assembler</i>	<i>low-level</i>
<i>program</i>	<i>instruction</i>	<i>machine</i>	<i>different</i>
<i>translation</i>	<i>manufacturer</i>	<i>language</i>	<i>version</i>

A compiler, in computer science, is a computer program that translates source code into object code. Software _____ write source code using high-level programming _____ that people can understand. Computers cannot _____ execute source code, but need a(n) _____ to translate these instructions into a(n) _____ language called machine code.

Compilers collect and reorganize (compile) all the _____ in a given set of source code to _____ object code. Object code is often the same as or _____ to a computer's machine code. If the object code is the same as the _____ language, the computer can run the _____ immediately after the compiler produces its _____. If the object code is not in machine language, other programs—such as _____, *binders* (联编程序), linkers, and *loaders* (装入程序)—finish the translation.

Most computer languages use different _____ of compilers for different types of _____ or operating systems, so one language may have _____ compilers for personal computers (PC) and Apple Macintosh computers. Many different _____ often produce versions of the same programming language, so compilers for a language may vary between manufacturers.

IV. Translate the following passage from English into Chinese:

In software, a bug is an error in coding or logic that causes a program to malfunction or to produce incorrect results. Minor bugs—for example, a cursor that does not behave as expected—can be inconvenient or frustrating, but not damaging to information. More severe bugs can cause a program to “hang” (stop responding to commands) and might leave the user with no alternative but to restart the program, losing any previous work that has not

been saved. In either case, the programmer must find and correct the error by the process known as debugging. Because of the potential risk to important data, commercial application programs are tested and debugged as completely as possible before release. Minor bugs found after the program becomes available are corrected in the next update; more severe bugs can sometimes be fixed with special software, called patches, that *circumvents* (规避) the problem or otherwise *alleviates* (减轻) its effects.

Section B

Model Driven Development

I. Introduction

modeling /'mɒdəlɪŋ/
n. 建模, 模型化

methodology
/ˌmeθə'dɒlədʒi/
n. (学科的) 一套方法; 方法论

building block
积木块, 构建模块, 构件

visualize
/'vɪʒʊəlaɪz/
v. 使可视化, 使可见; 使直观化

skip /skip/
v. 略过, 跳过

consolidate
/kən'solɪdeɪt/
v. (把...) 联为一体, 合并; 巩固

waterfall
/'wɔ:təfɔ:l/
n. 瀑布

omission /əu'mɪʃən/
n. 省略; 遗漏; 疏忽

One of the oldest and most commonly used approaches to analyzing and designing information systems is based on **modeling**.

Modeling is the act of drawing one or more graphical representations (or pictures) of a system. Modeling is a communication technique based on the old saying, "a picture is worth a thousand words."

In the FAST¹ **methodology**, system models are used to illustrate and communicate the DATA, PROCESS, or INTERFACE **building blocks** of information systems.² This approach is called model-driven development (MDD³).

Model-driven development techniques emphasize the drawing of models to help **visualize** and analyze problems, define business requirements, and design information systems.

Model-driven development is illustrated in Figure 4B-1. The model-driven approach usually does not **skip** or **consolidate** the basic phases. The model-driven route takes on the appearance of a **waterfall**, suggesting that phases must generally be completed in sequence.⁴ As shown in the figure, it is possible to back up to correct mistakes or **omissions**; however, such rework is often difficult, time consuming, and costly.⁵

¹ **FAST**: 系统技术应用框架 (Framework for the Application of Systems Techniques 的首字母缩略)。一种软件开发方法。

² *In the FAST methodology, system models are used to illustrate and communicate the DATA, PROCESS, or INTERFACE building blocks of information systems.*: 在 FAST 方法中, 系统模型用于信息系统的 DATA、PROCESS 或 INTERFACE 构建模块的表示 (描述) 和通信 (交互)。DATA、PROCESS 和 INTERFACE 指信息系统逻辑上的结构, 即 (存储) 数据层、(运算) 处理层和 (用户) 界面层, 是信息系统的构建模块。

³ **MDD**: 模型驱动开发 (model-driven development 的首字母缩略)。

⁴ *The model-driven route takes on the appearance of a waterfall, suggesting that phases must generally be completed in sequence.*: 模型驱动方法呈现出瀑布的样子, 也就是说各阶段一般必须按顺序完成。

⁵ *it is possible to back up to correct mistakes or omissions; however, such rework is often difficult, time consuming, and costly.*: 退回去更改错误和疏忽是能够做到的; 然而, 这种返工常常困难、耗时和费钱。

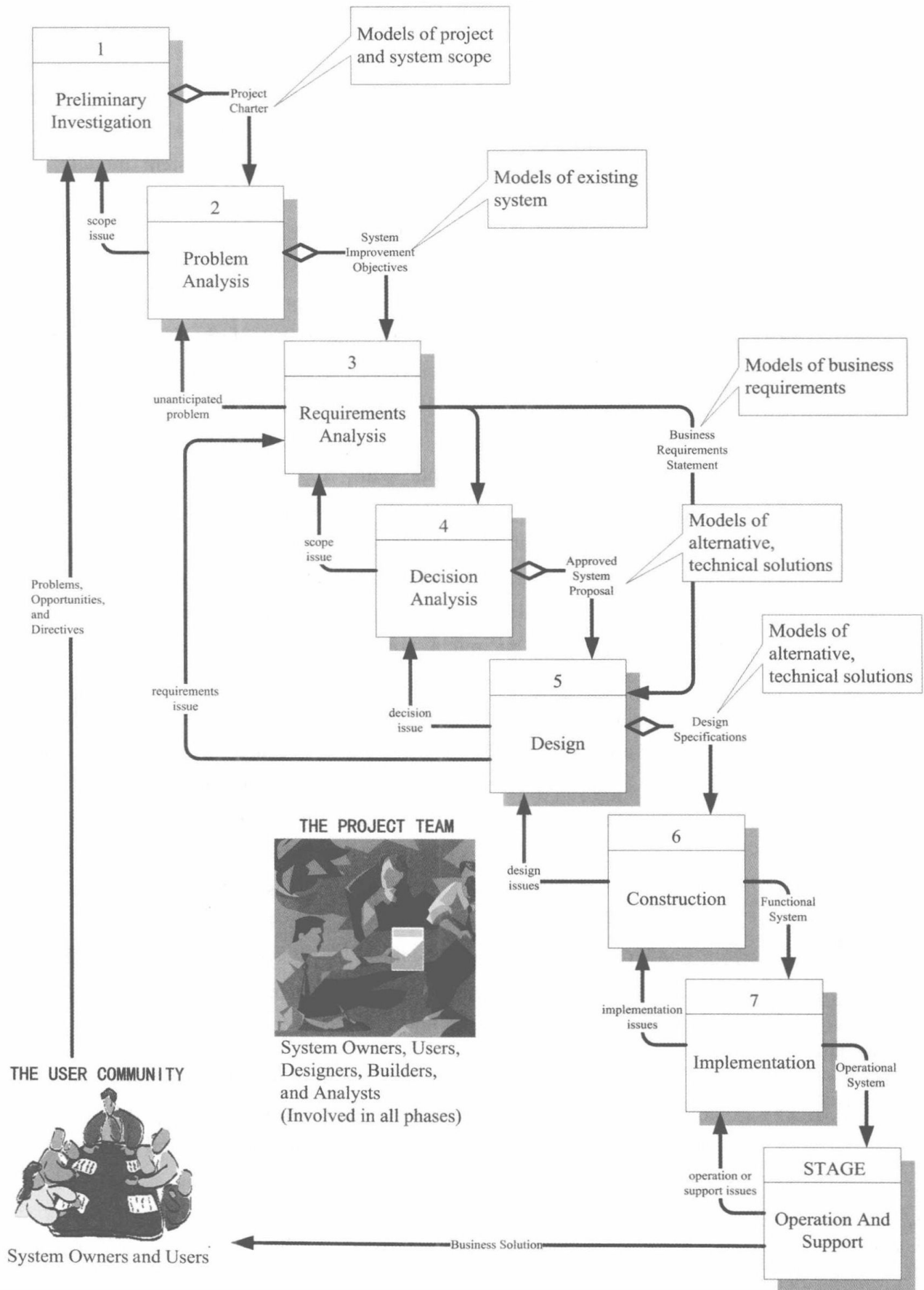


Figure 4B-1: Model-Driven Development Life Cycle

We call your attention to the following notes that correspond to Figure 4B-1.

premium

/ˈpri:miəm/

n. 奖品; 额外补贴;
加付款

bureaucratic

/ˌbjʊərəʊˈkrætɪk/

a. 官僚(政治)的;
官僚主义的

analyst /ˈænəlɪst/

n. 分析员, 分析师

structure chart

结构图

blueprint

/ˈblu:prɪnt/

n. 蓝图

deliverable

/dɪˈlɪvərəbəl/

n. [常作复] 可交付
使用的产品

documentation

/ˌdɒkjumənˈteɪʃən/

n. 文件编制, 文档
编制; [总称] 文件
证据, 文献资料

1. The model-driven approach places a **premium** on¹ planning because projects tend to be large and rework tends to be costly. During preliminary investigation, simple system models can be useful to visualize project scope. Defining scope is critical to estimating time and costs to complete the model-driven route.
2. Some system modeling techniques call for extensive modeling of the current system to identify problems and opportunities for improvement. This is especially true if the underlying business processes are believed to be inefficient or **bureaucratic**. This may also prove helpful if the system is very large and complex.
3. Most model-driven techniques require **analysts** to document business requirements with “logical” system models.² Logical models show only “what” a system must be or must do. They are implementation independent; that is, they depict the system independent of³ any possible technical implementation.
4. Many model-driven techniques require analysts to document alternative technical solutions with “physical” system models. Physical models show not only what a system is or does, but also how the system is physically and technically implemented. They are implementation dependent because they reflect technology choices and the limitations of those technology choices.
5. Detailed physical models are a common product of many system design techniques. Examples include database schemas, **structure charts**, and flowcharts. They serve as a **blueprint** for construction of the new system.
6. Construction translates the physical system models into software.⁴
7. During implementation, system models can be included in training and user manuals for the new system.

Essentially system models are produced as a portion of the **deliverables** for each phase. Once implemented, the models serve as **documentation** for any changes that might be needed during the operation and support stage of

¹ *place a (high) premium on*: 亦作 *put a (high) premium on*, 意为“高度评价”、“高度重视”。

² *to document business requirements with “logical” system models*: 用“逻辑”系统模型编写业务需求文档。在这里, *document* 作动词, 意为“编制…的文档”、“对…进行文档化”。

³ *independent of*: 独立于…以外的; 不管, 不顾。

⁴ *Construction translates the physical system models into software.*: 系统构造将系统物理模型转变成软件。

the life cycle.

The model-driven approach offers several advantages:

- It minimizes planning overhead because all the phases are planned up front.¹ (This does not mean the project cannot become infeasible and get canceled.)
- Requirements analysis tends to be more thorough and better documented in the model-driven approach.
- Alternative technical solutions tend to be more thoroughly analyzed in the model-driven approach.
- System designs tend to be more sound, stable, adaptable, and flexible because they are model-based and more thoroughly analyzed before they are built.
- The approach is effective for systems that are well understood but so complex that they require large project teams to complete.
- The approach works well when fulfilling user expectations and quality are more important than cost and schedule.

There are also several disadvantages of model-driven development. Most often cited is the long duration of projects; it takes time to collect the facts, draw the models, and **validate** those models. This is especially true if users are uncertain or imprecise about their system requirements. The models can only be as good as the users' understanding of those requirements. Second, pictures are not software; some argue that this reduces the users' role in a project to passive participation. Most users don't get excited about pictures. Instead, they want to see working software and they **gauge** project progress by the existence of software (or its absence). Finally, the model-driven approach is considered by some to be inflexible; users must fully specify requirements before design; design must fully document technical specifications before construction; and so forth. Some view such **rigidity** as impractical. Regardless, model-driven approaches remain popular.

There are several different model-driven techniques. They differ primarily in terms of the types of models that they require the analyst to draw and validate.

Let's briefly examine three common model-driven development techniques. Please note that we are only introducing the techniques here,

validate /ˈvælideɪt/
v. 确认 (…有效),
证实, 验证

gauge /geɪdʒ/
v. 估计, 判断;
计量

rigidity /rɪˈdʒɪdɪti/
n. 严格; 刻板

¹ *It minimizes planning overhead because all the phases are planned up front.*: 它将计划费用降到最低, 因为所有阶段都是预先计划好的。在这句话中, up front 意为“预先”。

not the models.

II. Structured Analysis and Design

Structured analysis and design were two of the first formal system modeling techniques.¹

data flow diagram
数据流程图

Structured analysis is a PROCESS-centered technique that is used to model business requirements for a system. Structured analysis introduced a modeling tool called the **data flow diagram**, used to illustrate business process requirements.

Structured design is a PROCESS-centered technique that transforms the structured analysis models into good software design models. Structured design introduced a modeling tool called structure charts, used to illustrate software structure to fulfill business requirements.

The information system building blocks include several possible focuses: data, processes, and interfaces. Structured analysis and design tend to focus primarily on the process building blocks. Thus, data flow diagrams and structure charts focus on business and software processes in the system. (The techniques have evolved to also consider the data building blocks as a secondary emphasis.)

Data flow diagrams and structure charts contributed significantly to reducing the communication gap that often exists between the nontechnical system owners and users and the technical system designers and builders.² For this reason, structured analysis and design are still among the most widely practiced modeling techniques.

III. Information Engineering (IE)

Many organizations have chosen information engineering modeling techniques.

entity /'entiti/

n. 实体

entity relationship diagram

实体关系图, 实体联系图, E-R 图

Information engineering is a DATA-centered but PROCESS-sensitive technique used to model business requirements and design systems that fulfill those requirements. Information engineering emphasizes a modeling tool called **entity relationship diagrams** to model business requirements.

¹ *Structured analysis and design were two of the first formal system modeling techniques.*: 结构化分析与设计是最早的正规的系统建模技术中的两种。

² *the communication gap that often exists between the nontechnical system owners and users and the technical system designers and builders*: 不懂技术的系统拥有者和用户, 与懂技术的系统设计者和构建者之间, 在交流上常常存在的隔阂。

With respect to the information system building blocks, information engineering tends to focus primarily on the data building blocks. Thus, entity relationship diagrams focus on data required in the system. Information engineering subsequently uses data flow diagrams and structure charts to model the system's processes that capture, store, and use that data.

Technically, information engineering is more ambitious than we have described here. At the risk of oversimplification, information engineering attempts to use its techniques on the organization as a whole—modeling the organization's entire data requirements, as opposed to a single information system in the organization.

Regardless, the DATA-centered technique in information engineering remains a practical option for developing individual information systems.

IV. Object-Oriented Analysis and Design (OOAD)

Object-oriented analysis and design are the new kids on the block.¹ But a simplified introduction is appropriate here.

For the past 30 years, techniques such as structured analysis and design and information engineering have deliberately separated concerns of data from those of processes. In other words, data and process models are separate and distinct. But these concerns must be synchronized. Object techniques are an attempt to eliminate the separation of concerns about data and process.

Object-oriented analysis and design attempt to merge the data and process concerns into singular constructs called objects.² Object-oriented analysis and design introduced **object diagrams** that document a system in terms of its objects and their interactions.

Business objects might correspond to real things of importance in the business such as customers and the orders they place for products. Each object consists of both the data that describes the object and the processes that can create, read, update, and delete that object.

With respect to the information system building blocks, object-oriented analysis and design significantly change the **paradigm**. The data and process columns are essentially merged into a single object column. The

object diagram
对象图

paradigm
/'pærədaɪm/
n. 范例; 范式

¹ *Object-oriented analysis and design are the new kids on the block.*: 面向对象分析与设计是这个街区新来的孩子 (亦即新出现的模型驱动开发技术)。

² *Object-oriented analysis and design attempt to merge the data and process concerns into singular constructs called objects.*: 面向对象分析与设计试图将与数据和处理过程有关的事务融合到称为对象的单一结构中。

frame / freim /

n. 图文框; 框架

drop-down menu

下拉式菜单 (或选项屏)

radio button

单选 (按) 钮, 小圆按钮

check box

复选框, 选择框, 校验框

scroll / skrɔl /

v. & n. 滚动

scroll bar

滚动条

maximize

/'mæksimaiz /

v. 使增加 (或扩大) 到最大限度

resize / ri:'saiz /

v. 调整大小, 重定尺寸

models then focus on identifying objects, building objects, and assembling appropriate objects into useful information systems.

Object models are also extendable to the interface building blocks of the information system framework. Most contemporary computer user interfaces are already based on object technology. For example, both the Microsoft Windows and Netscape Navigator¹ interfaces use standard objects such as windows, frames, drop-down menus, radio buttons, check boxes, scroll bars, and the like. Consider a window object. It can be characterized in terms of data such as height, width, color, and so on. And it has methods such as minimizing, maximizing, resizing, and so forth. Object programming technologies such as C++, Java, Smalltalk, and Visual Basic² are used to construct and assemble such interface objects.

Exercises

I. Fill in the blanks with the information given in the text:

1. Information engineering emphasizes a modeling tool called _____ relationship diagrams.
2. One of the disadvantages of model-driven development is the long _____ of projects.
3. Unlike structured analysis and design and information engineering, object-oriented analysis and design attempt to merge the _____ and _____ concerns into singular constructs called objects.
4. Unlike logical models, physical models show not only what a system is or does, but also how the system is physically and technically _____.

II. Translate the following terms or phrases from English into Chinese and vice versa:

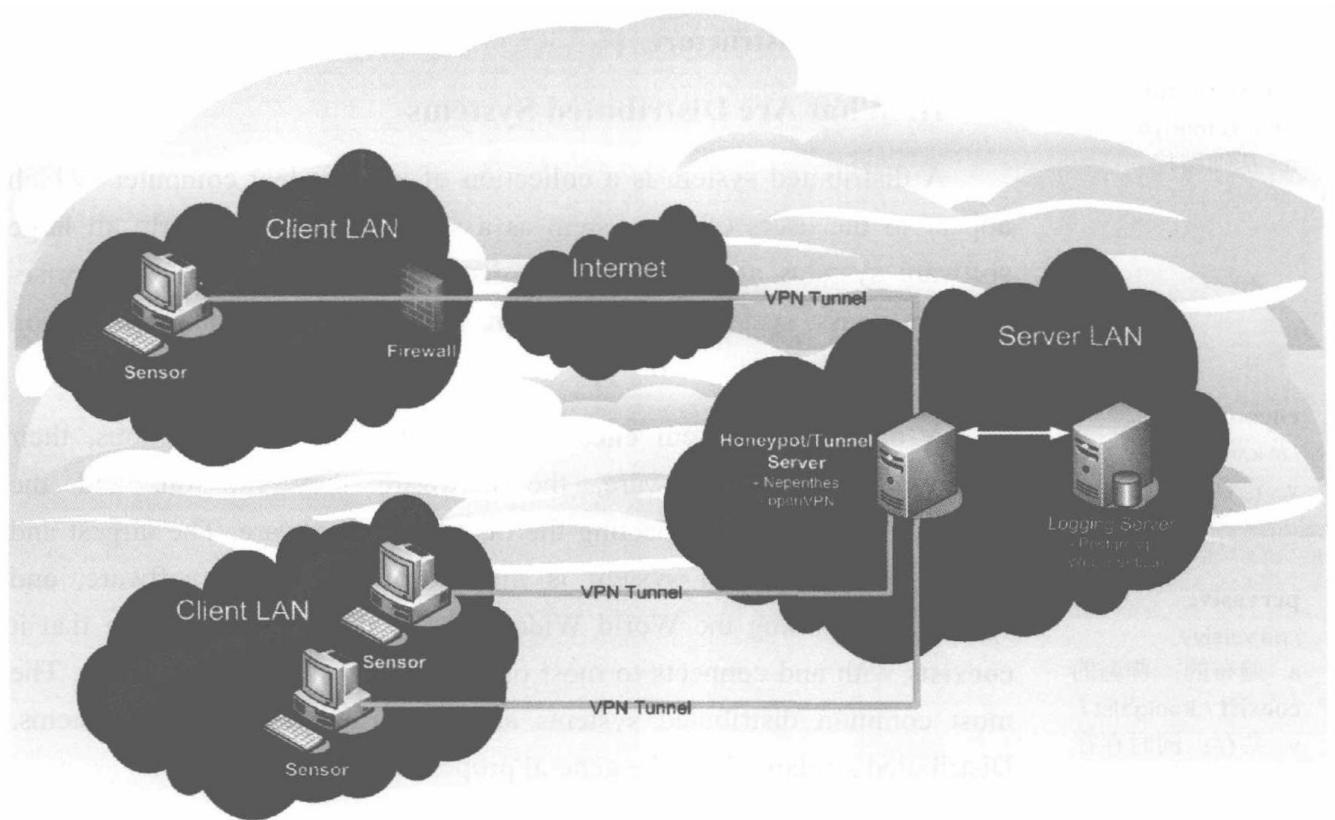
- | | |
|----------------------|-----------|
| 1. check box | 6. 系统建模技术 |
| 2. structured design | 7. 模型驱动开发 |
| 3. building block | 8. 数据流程图 |
| 4. database schema | 9. 下拉式菜单 |
| 5. radio button | 10. 滚动条 |

¹ Netscape Navigator: 网景浏览器, 美国网景 (Netscape) 公司开发的网络浏览器。

² Visual Basic: 可视化 Basic 语言, 简称 VB, 系微软公司开发的一种事件驱动编程语言和集成开发环境 (IDE), 最初于 1991 年发行。

Section C

Distributed Systems



I. Introduction

The field of distributed computing has witnessed an explosive expansion during the last decade. As the use of distributed computing systems for large-scale computations is growing, so is the need to increase their reliability. Nevertheless, the probability of failure of an individual processing **node** in multinode distributed systems is not **negligible**. Hence, it is necessary to develop mechanisms that prevent the waste of computations performed on distributed processing nodes if one of the nodes fails, either due to a hardware **transient** fault (bus error or segmentation fault) or a permanent fault (power failure or communication network malfunction).

Advances in communications technology and methods of work introduced at diverse workplace environments naturally led to a greater

node /nəʊd/

n. 节点

negligible

/ˈneglɪdʒəbl/

a. 可略而不计的

transient

/ˈtrænzɪənt/

a. 短暂的; 临时的

segmentation

/ˌsegmenˈteɪʃən/

n. (程序) 分段, 分割

homogeneous

/ˌhɒməʊdʒiːniəs/

a. 同种类的; 同性质的; 同构的

heterogeneous

/ˌhetərəʊdʒiːniəs/

a. 不同种类的, 多机种的, 异构的

infrastructure

/ˈɪnfəˌstrʌktʃə/

n. 基础结构

encompass

/ɪnˈkʌmpəs/

v. 包含, 包括

pervasive

/pəˈveɪsɪv/

a. 遍布的; 普遍的

coexist /ˌkəʊɪɡˈzɪst/

v. 共存; 同时存在

stock quote

股票行情

collaborative

/kəˈlæbəreɪtɪv/

a. 合作的, 协作的

distribution of information processing. Initially, most distributed systems were **homogeneous**, but now many distributed environments are **heterogeneous**. Therefore, the distributed systems design must focus on **heterogeneous** environments, treating homogeneous systems as special cases in a heterogeneous world. Key issues in distributed systems design include where specific functionality should be located within the information **infrastructure**.

II. What Are Distributed Systems

A distributed system is a collection of independent computers which appear to the users of the system as a single computer. Nearly all large software systems are by necessity¹ distributed. For example, enterprise-wide business systems must support multiple users running common applications across different sites.

A distributed system **encompasses** a variety of applications, their underlying support software, the hardware they run on, and the communication links connecting the distributed hardware. The largest and best-known distributed system is the set of computers, software, and services comprising the World Wide Web², which is so **pervasive** that it **coexists** with and connects to most other existing distributed systems. The most common distributed systems are networked client/server systems. Distributed systems share the general properties described below.

1. Resource Sharing

The most common reason for connecting a set of computers into a distributed system is to allow them to share physical and computational resources (printers, files, databases, mail services, **stock quotes**, and **collaborative** applications, for example). Distributed system components that support resource sharing play a similar role as, and are increasingly indistinguishable from, operating systems.

2. Multiple Nodes

Software for the distributed system executes on nodes, or multiple independent computers (not merely multiple processors on the same computer, which is the realm of parallel computing). These nodes can range among personal computers, high-performance workstations, file servers,

¹ *by necessity*: 无法避免地, 势必, 必定。

² *World Wide Web*: 万维网 (亦作 World-Wide Web, 常缩略为 WWW、w3、W3 和 Web)。

peer /piə/
n. 同级设备; 同级, 同层, 对等层
connectivity
/kənek'tiviti/
n. 连通(性), 连接(性)

mainframes, and supercomputers. Each can take the role of a client, which requests services by others; a server, which provides computation or resource access to others; or a peer, which does both. A distributed system may be as small as two nodes, provided software **connectivity** is present. This arrangement is represented in Figure 4C-1.

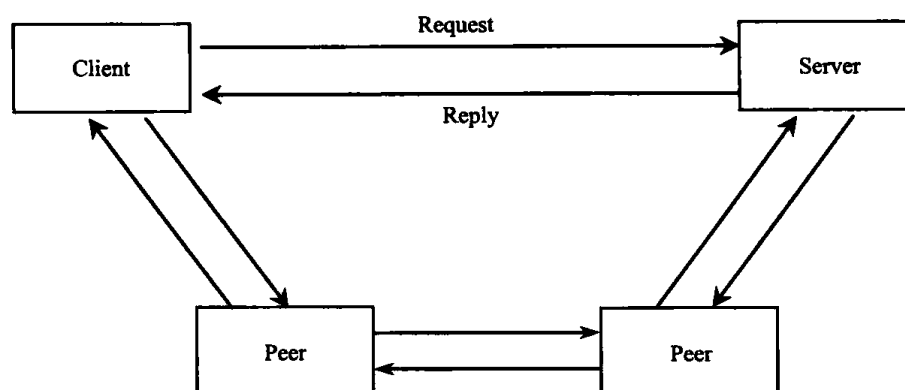


Figure 4C-1: A Small Distributed System

concurrency
/kən'kʌrənsi/
n. 同时发生, 并发, 并行性

3. Concurrency

Each of the nodes in a distributed system functions both independently and concurrently with all of the others. More than one process (executing program) per node and more than one thread (concurrently executing task) per process can act as components in a system. Most components are reactive, continuously responding to commands from users and messages from other components. Like operating systems, distributed systems are designed to avoid **termination** and so should always remain at least partially available.

termination
/ˌtɜːmɪ'neɪʃən/
n. 结束, 终止

4. Heterogeneity

heterogeneity
/ˌhetərəʊdʒi'niːəti/
n. 各种各样, 异构性, 非均匀性

The nodes participating in a system can consist of diverse computing and communication hardware. The software comprising the system can include diverse programming languages and development tools. Some heterogeneity issues can be addressed with common message formats and low-level protocols that are readily implemented across different platforms (e.g., PCs, servers, and mainframes). Others may require construction of **bridges** that translate one set of formats and protocols to another. More thorough **system integration** can be attained by requiring that all nodes support a common **virtual machine** that processes platform-independent program instructions. The systems that use the Java programming language

bridge /brɪdʒ/
n. 网桥, 桥接器
system integration
系统集成
virtual machine
虚拟机, 虚(拟计算)机

follow this approach.

5. Multiple Protocols

Most distributed message passing differs significantly from the kinds of **invocations** (such as procedure calls) used within the confines of **sequential** programs. The most basic form of distributed communication is **asynchronous**. Similar to letters mailed in a postal system, senders issue messages without relying on receipt of or reply by their **recipients**. Such basic distributed messages usually take much longer to reach recipients than do local invocations. They sometimes reach recipients in a different order than they were sent and they may fail to reach them at all. To avoid this, more sophisticated protocols must be constructed. These may include:

- **Procedural** messaging, in which senders wait for full replies
- **Semi-synchronous** messaging, in which senders wait for an acknowledgment of message receipt before proceeding
- **Transactional** protocols, in which all messages in a given **session** or transaction are processed in an all-or-none fashion¹
- **Callback** protocols, in which receivers later issue different messages back to their senders
- **Time-out** protocols, in which senders only wait for replies for a certain period before proceeding
- **Multicast** protocols, in which senders simultaneously issue messages to a group of other nodes

These and other protocols are often extended and specialized to enhance reliability, security, and efficiency.

6. Fault Tolerance

A program running on a single computer is, at best, only as reliable as that computer. Most distributed systems, on the other hand, need to remain at least partially available and functional even if some of their nodes, applications, or communication links fail or misbehave. In addition to **outright** failures, applications may suffer from unacceptably low quality of service due to **bandwidth** shortages, network **contention**, **software overhead**, or other system limitations, so fault-tolerance requirements present some of the most central, yet difficult challenges in the construction

invocation

/,invə'keɪʃən/

n. 调用

sequential

/si'kwɛnʃəl/

a. 顺序的

asynchronous

/ei'sɪŋkrənəs/

a. 不同时的; 异步的, 非同步的

recipient

/ri'sɪpiənt/

n. 接受者; 收件人

synchronous

/'sɪŋkrənəs/

a. 同时的; 同步的

session

/'seʃən/

n. 对话(期), 会话

multicast

/'mʌltɪkɑːst/

n. 组播(通信), 多址通信

multicast protocol

多址通信协议

fault tolerance

容错

outright

/'aʊtraɪt/

a. 完全的, 彻底的

bandwidth

/'bændwɪð/

n. 带宽

contention

/kən'tenʃən/

n. 争用; 争夺

software overhead

软件开销

¹ *Transactional protocols, in which all messages in a given session or transaction are processed in an all-or-none fashion:* 事务处理协议, 即在一个给定的会话或事务处理内, 所有消息要么被全部处理, 要么全部不被处理。

of distributed systems.

authorized

/ˈɔːθəraɪzd/

a. 经授权的, 特许的

intrinsically

/ɪnˈtrɪnsɪkəli/

ad. 固有地; 本质上; 内在地

encryption

/ɪnˈkriptʃən/

n. 加密

authentication

/ɔːθəntɪˈkeɪʃən/

n. 验证, 鉴别

trustworthiness

/ˈtrʌst,wəːðɪnɪs/

n. 值得信任; 可信; 可靠

Ethernet

/ˈiːθə.net/

n. 以太网 (标准)

mediate

/ˈmiːdiət/

v. 调解; 传递

routing

/ˈruːtɪŋ/

n. 路由选择

routing algorithm

路由算法

notification

/ˌnəʊtɪfɪˈkeɪʃən/

n. 通知

multimedia

/ˌmʌltiˈmiːdiə/

a. (使用) 多媒体的

extensibility

/ɪkˌstenseɪbɪlɪti/

n. 可扩展性, 可扩充性

accommodate

/əˈkɒmədeɪt/

v. 容纳; 使适应

7. Security

Only **authorized** users may access sensitive data or perform critical operations. Security in distributed systems is **intrinsically** a multilevel issue, ranging from the basic safety guarantees provided by the hardware and operating systems residing on each node; to message **encryption** and **authentication** protocols; to mechanisms supporting issues concerning privacy, appropriateness of content, and individual responsibility.

Techniques for addressing **trustworthiness** include using digital certificates and preventing component code performing potentially dangerous operations such as modifying disk files.

8. Message Passing

Software on separate computers communicates via structured message-passing disciplines built upon a number of networking protocols (for example, TCP/IP¹). These, in turn, may run on any of a number of connection technologies (for example, **Ethernet** and modems). The nodes in most distributed systems are completely connected—any node may send a message to any other node. Delivery is **mediated** by underlying **routing algorithms** and related networking support.

Messages include commands, requests for services, event **notifications**, **multimedia** data, file contents, and even entire programs. It should be noted that most multiprocessors communicate via shared memory rather than message passing and therefore are not distributed.

9. Openness

Most sequential programs are considered closed because their configurations never change after execution commences. Most distributed systems are, to some degree, open, because nodes, components, and applications can be added or changed while the system is running. This provides the **extensibility** necessary to **accommodate** expansion, and the ability to evolve and cope with the changing world in which a system resides.

Openness requires that each component obey a certain minimal set of

¹ **TCP/IP**: TCP/IP 协议, 传输控制 (协议) / 网际 (或网间) 协议 (Transmission Control Protocol/Internet Protocol 的首字母缩略)。

interoperability

/ˈɪntəˌɒpəreɪbɪlɪti/

n. 互操作性, 互用性

hypertext

/ˈhaɪpetekst/

n. 超(级)文本

Hypertext Transfer Protocol 超(级)文本传送(或传输)协议**consortium**

/kənˈsɔːtɪəm/ ([复])

-tia / -tiə / 或 -tiums)

n. 会, 社; (国际性的) 财团

-specific /spiˈsɪfɪk/

comb. form 表示“限定的”, “特有的”

domain /deɪˈmeɪn/

n. 领域, 域

autonomous

/ɔːˈtɒnəməs/

a. 自治的, 自主的, 独立的

corporate

/ˈkɔːpəɪt/

a. 公司的; 社团的

firewall /ˈfaɪəwɔːl/

n. 防火墙

intranet /ˈɪntreɪnɪt/

n. 内部网, 企业内部网, 内联网, 内网

persistence

/pəˈsɪstəns/

n. 持续性, 持久性

persistent

/pəˈsɪstənt/

a. 持续的, 持久的

outlast

/ˌaʊtˈlɑːst; -ləst/

v. 比...持久

autonomous agent

自主主体

aggregate

/ˈægrɪgeɪt/

a. 聚集的; 合计的

policies, conventions, and protocols to ensure **interoperability** among updated or added components. Historically, the most successful open systems have been those with the most minimal requirements. For example, the simplicity of the **Hypertext Transfer Protocol (HTTP)** was a major factor in the success of the World Wide Web.

Standards organizations such as the International Organization for Standardization (ISO¹) and American National Standards Institute (ANSI²), along with industrial **consortia** such as the Object Management Group (OMG³), establish the basic format and protocol standards underlying many interoperability guarantees. Individual distributed systems additionally rely on **context-specific** or **domain-dependent** policies and mechanisms.⁴

10. Isolation

Each component is logically or physically **autonomous**, and it communicates with others only via structured message protocols. In addition, groups of components may be segregated for purposes of functionality, performance, or security. For example, while the connectivity of a **corporate** distributed system might extend to the entire Internet, its essential functionality could be segregated (often by a **firewall**) to an **intranet** operating only within the firewall. It would communicate then with other parts of the system via a restricted secure protocol.

11. Persistence

At least some data and programs are maintained on **persistent** media that **outlast** the execution of a given application. Persistence may be arranged at the level of file systems, database systems, or programming language runtime support mechanisms.

12. Decentralized Control

No single computer is necessarily responsible for configuration, management, or policy control for the system as a whole. Distributed systems are instead domains joined by protocol of **autonomous agents** that agree on enough common policies to provide an **aggregate** functionality. Some aspects of decentralization are desirable, such as fault-tolerance

¹ ISO: 国际标准化组织 (International Organization for Standardization 的首字母缩略)。

² ANSI: 美国国家标准协会 (American National Standards Institute 的首字母缩略)。

³ OMG: 对象管理组 (Object Management Group 的首字母缩略)。

⁴ Individual distributed systems additionally rely on context-specific or domain-dependent policies and mechanisms.: 单个分布式系统还依赖于和上下文细节或域相关的策略和机制。

administer

/əd'ministə/

v. 掌管; 实施

provisions. Others are essential because centralized control cannot accommodate the number of nodes and connections supported by contemporary systems. The tools for **administering** system-wide policies, however, may be restricted to particular users.

III. Advantages and Disadvantages

1. Advantages of Distributed Systems

Distributed systems have many inherent advantages, especially over centralized systems. Some applications are inherently distributed as well. In general, distributed systems:

- Yield higher performance
- Provide higher reliability
- Allow incremental growth. Distributed computing offers higher rates of return over individual CPUs:
 - It is both feasible and easy to construct systems of a large number of CPUs connected by a high-speed network.
 - It answers a need to share data scattered over these CPUs.
 - It provides a way to share expensive **peripherals**.
 - It allows one user to run a program on many different machines.

peripheral

/pə'riferəl/

n. 外围设备,
外部设备

2. Disadvantages of Distributed Systems

In spite of their many advantages, distributed systems do create a few disadvantages. Some of these are:

- The need for new operating systems to support them
- A **reliance** on network communications
- The need for enhanced security
- Offer no nice classification of operating systems
- Use **loosely** and **tightly coupled systems**

reliance /ri'laiəns/

n. 依靠, 依赖

**loosely coupled
system**

松散耦合系统

**tightly coupled
system**

紧密耦合系统

cost effectiveness

成本效益

IV. Conclusions

We want to fully utilize a heterogeneous computing environment where different types of processing resources and interconnection technologies are used effectively and efficiently. Using distributed resources provides the potential of maximizing performance and **cost effectiveness** for a wide range of scientific and distributed applications.

Distributed computing environments comprising networked heterogeneous

workstations are becoming the standard configuration in both engineering and scientific environments. However, the lack of a unifying parallel computing model (a parallel equivalent of a von Neumann model) means that the current parallel applications are nonportable.

Exercises

I. Fill in the blanks with the information given in the text:

1. Now many distributed systems are _____ whereas most of them were initially homogeneous.
2. The most basic form of distributed communication is _____.
3. Some of the heterogeneity issues in a system may be dealt with through the construction of _____ that translate one set of formats and protocols to another.
4. One of the general properties shared by distributed systems is _____, which allows nodes, components, and applications to be added or changed while the systems are running.

II. Translate the following terms or phrases from English into Chinese and vice versa:

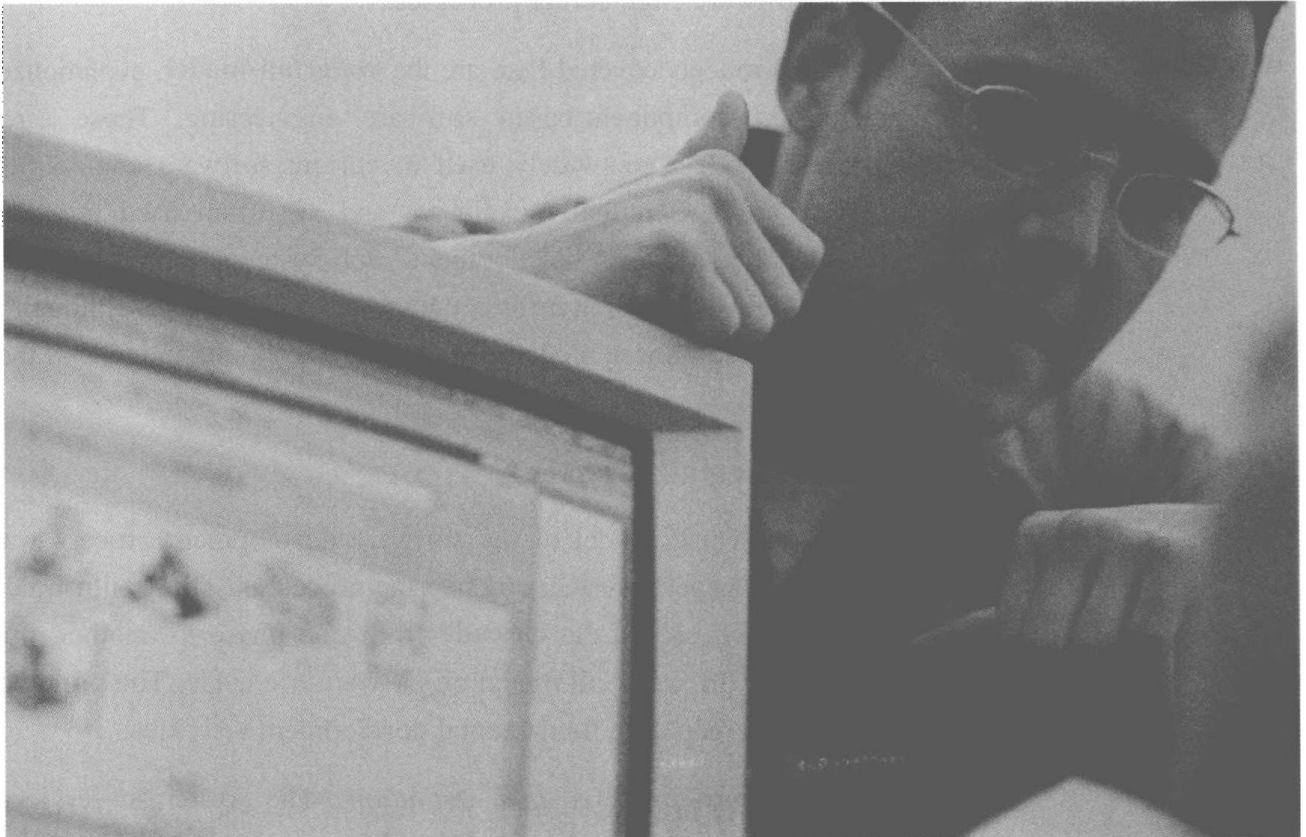
- | | |
|-----------------------|-----------|
| 1. procedure call | 5. 路由算法 |
| 2. fault tolerance | 6. 异构型环境 |
| 3. homogeneous system | 7. 多址通信协议 |
| 4. autonomous agent | 8. 通信链路 |

Unit 5 Software Process

(软件过程)

Section A

Software Process Models



I. Introduction

A software process is a set of activities that leads to the production of a software product. These activities may involve the development of software from scratch in a standard programming language like Java or C. Increasingly, however, new software is developed by extending and modifying existing systems and by configuring and integrating **off-the-shelf** software or system components.

off-the-shelf

/'ɒfðə'self /

a. 现成的，非专门设计（或定制）的

A software process model is an abstract representation of a software process. Each process model represents a process from a particular

perspective, and thus provides only partial information about that process. This section introduces a number of very general process models (sometimes called *process paradigms*) and presents them from an architectural perspective. That is, we see the framework of the process but not the details of specific activities.

definitive / di'finitiv /

a. 决定性的; 确定的; 规定的

abstraction

/ æb'strækʃən /

n. 抽象; 提取

These generic models are not **definitive** descriptions of software processes. Rather, they are **abstractions** of the process that can be used to explain different approaches to software development. You can think of them as process frameworks that may be extended and adapted to create more specific software engineering processes.

waterfall model

瀑布模型

The process models covered here are the **waterfall model**, evolutionary development and component-based software engineering. These three generic process models are widely used in current software engineering practice. They are not mutually exclusive and are often used together, especially for large systems development. Sub-systems within a larger system may be developed using different approaches. Therefore, although it is convenient to discuss these models separately, you should understand that, in practice, they are often combined.

II. The Waterfall Model

The first published model of the software development process was derived from more general system engineering processes. This is illustrated in Figure 5A-1. Because of the **cascade** from one phase to another, this model is known as the waterfall model or software life cycle. The principal stages of the model map onto fundamental development activities:

cascade / kæ'skeid /

n. 小瀑布; 瀑布状物; 级联; 层叠

constraint

/ kən'streint /

n. 约束, 限制

partition / pɑ:'tɪʃən /

v. 划分, 分区, 分割

unit testing

单元测试

program unit

程序单元

1. *Requirements analysis and definition.* The system's services, **constraints** and goals are established by consultation with system users. They are then defined in detail and serve as a system specification.
2. *System and software design.* The systems design process **partitions** the requirements to either hardware or software systems. It establishes an overall system architecture. Software design involves identifying and describing the fundamental software system abstractions and their relationships.
3. *Implementation and unit testing.* During this stage, the software design is realized as a set of programs or **program units**. Unit

testing involves verifying that each unit meets its specification.

4. *Integration and system testing.* The individual program units or programs are integrated and tested as a complete system to ensure that the software requirements have been met. After testing, the software system is delivered to the customer.

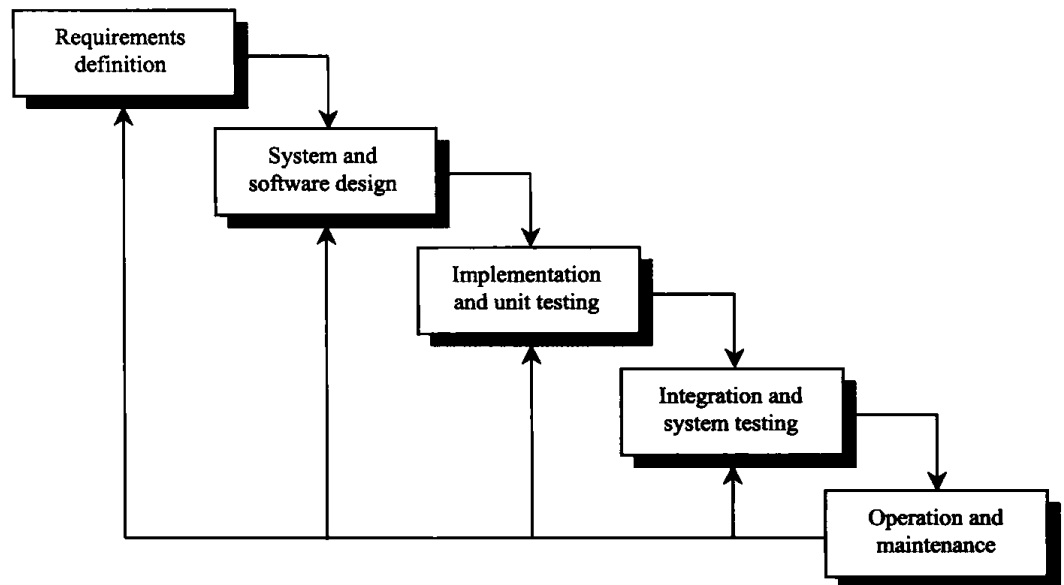


Figure 5A-1: The Software Life Cycle

5. *Operation and maintenance.* Normally (although not necessarily) this is the longest life-cycle phase. The system is installed and put into practical use. Maintenance involves correcting errors which were not discovered in earlier stages of the life cycle, improving the implementation of system units and enhancing the system's services as new requirements are discovered.

In principle, the result of each phase is one or more documents that are approved. The following phase should not start until the previous phase has finished. In practice, these stages overlap and feed information to each other. During design, problems with requirements are identified; during coding design problems are found and so on. The software process is not a simple linear model but involves a sequence of **iterations** of the development activities.

iteration

/ˌɪtə'reɪʃən/

n. 迭代 (法),
重复

Because of the costs of producing and approving documents, iterations are costly and involve significant rework. Therefore, after a small number of iterations, it is normal to freeze parts of the development, such as the specification, and to continue with the later development stages. Problems

premature

/.premə'tjuə; ,pri:-/

a. 不成熟的, 仓促的, 过早的

circumvent

/.sə:kəm'vent/

v. 绕过; 规避

are left for later resolution, ignored or programmed around.¹ This premature freezing of requirements may mean that the system won't do what the user wants. It may also lead to badly structured systems as design problems are circumvented by implementation tricks.

During the final life-cycle phase (operation and maintenance), the software is put into use. Errors and omissions in the original software requirements are discovered. Program and design errors emerge and the need for new functionality is identified. The system must therefore evolve to remain useful. Making these changes (software maintenance) may involve repeating previous process stages.

The advantages of the waterfall model are that documentation is produced at each phase and that it fits with other engineering process models. Its major problem is its inflexible partitioning of the project into distinct stages. Commitments must be made at an early stage in the process, which makes it difficult to respond to changing customer requirements.

Therefore, the waterfall model should only be used when the requirements are well understood and unlikely to change radically during system development. However, the waterfall model reflects the type of process model used in other engineering projects. Consequently, software processes based on this approach are still used for software development, particularly when the software project is part of a larger systems engineering project.

III. Evolutionary Development

Evolutionary development is based on the idea of developing an initial implementation, exposing this to user comment and refining it through many versions until an adequate system has been developed (Figure 5A-2). Specification, development and validation activities are interleaved rather than separate, with rapid feedback across activities.

There are two fundamental types of evolutionary development:

1. *Exploratory development* where the objective of the process is to work with the customers to explore their requirements and deliver a final system. The development starts with the parts of the system that are understood. The system evolves by adding new features

validation

/.væli'deɪʃən/

n. 确认, 验证

interleave /'inteli:v/

v. 交错, 交叉, 交替; 隔行扫描

exploratory

/ik'splɒrətəri/

a. 探索的; 勘探的

¹ Problems are left for later resolution, ignored or programmed around.: 问题留待以后解决, 置之不顾, 或者通过编程绕过去。

proposed by the customer.

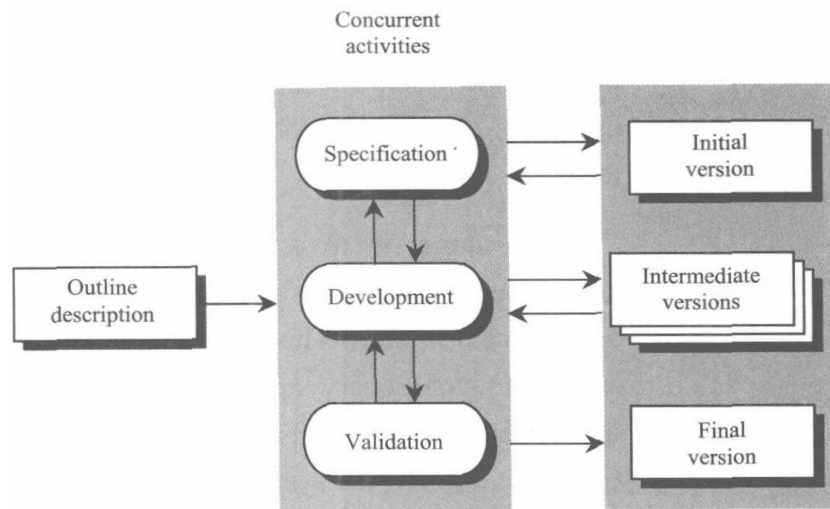


Figure 5A-2: Evolutionary Development

throwaway

/ˈθrəʊə,wei/

a. 使用后抛弃的，
一次性使用的

prototyping

/ˈprəʊtəˌtaɪpɪŋ/

n. 原型法；原型开发（或设计）；样机研究

2. **Throwaway prototyping** where the objective of the evolutionary development process is to understand the customer's requirements and hence develop a better requirements definition for the system. The prototype concentrates on experimenting with the customer requirements that are poorly understood.

An evolutionary approach to software development is often more effective than the waterfall approach in producing systems that meet the immediate needs of customers. The advantage of a software process that is based on an evolutionary approach is that the specification can be developed incrementally. As users develop a better understanding of their problem, this can be reflected in the software system. However, from an engineering and management perspective, the evolutionary approach has two problems:

1. *The process is not visible.* Managers need regular deliverables to measure progress. If systems are developed quickly, it is not **cost-effective** to produce documents that reflect every version of the system.
2. *Systems are often poorly structured.* Continual change tends to **corrupt** the software structure. Incorporating software changes becomes increasingly difficult and costly.

For small and medium-sized systems (up to 500,000 lines of code), the

cost-effective

/ˈkɒstiˈfektɪv/

a. 有成本效益的；
合算的

corrupt

/kəˈrʌpt/

v. 破坏，损坏；
腐蚀

evolutionary approach may be the best approach to development. The problems of evolutionary development become particularly acute for large, complex, long-life-time systems, where different teams develop different parts of the system. It is difficult to establish a stable system architecture using this approach, which makes it hard to integrate contributions from the teams.

For large systems, a mixed process is recommended which incorporates the best features of the waterfall and the evolutionary development models. This may involve developing a throwaway prototype using an evolutionary approach to resolve uncertainties in the system specification. You can then reimplement the system using a more structured approach. Parts of the system that are well understood can be specified and developed using a waterfall-based process. Other parts of the system, such as the user interface, which are difficult to specify in advance, should always be developed using an exploratory programming approach.

IV. Component-Based Software Engineering

In the majority of software projects, there is some software reuse. This usually happens informally when people working on the project know of designs or code which is similar to that required. They look for these, modify them as needed and incorporate them into their system. In the evolutionary approach, reuse is often essential for rapid system development.

irrespective
/iri'spektiv/
a. 不考虑的;
不顾的 (of)

This informal reuse takes place **irrespective** of the development process that is used. However, in the last few years, an approach to software development called component-based software engineering (CBSE¹), which relies on reuse, has emerged and is becoming increasingly used.

This reuse-oriented approach relies on a large base of reusable software components and some integrating framework for these components. Sometimes, these components are systems in their own right (COTS² or commercial off-the-shelf systems) that may provide specific functionality such as text formatting or numeric calculation. The generic process model for CBSE is shown in Figure 5A-3.

While the initial requirements specification stage and the validation stage are comparable with other processes, the intermediate stages in a

¹ CBSE: 基于组件的软件工程 (component-based software engineering 的首字母缩略)。

² COTS: 现成商用的, 现成民用的 (commercial off-the-shelf 的首字母缩略)。

reuse-oriented process are different. These stages are:

1. *Component analysis*. Given the requirements specification,¹ a search is made for components to implement that specification. Usually, there is no exact match, and the components that may be used only provide some of the functionality required.

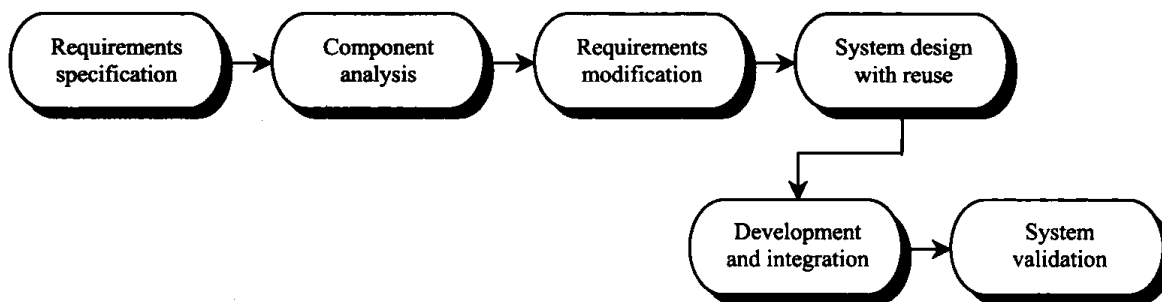


Figure 5A-3: Component-Based Software Engineering

modification

/ˌmɒdɪfɪˈkeɪʃən/

n. 修改, 更改

2. *Requirements modification*. During this stage, the requirements are analyzed using information about the components that have been discovered. They are then modified to reflect the available components. Where modifications are impossible, the component analysis activity may be re-entered to search for alternative solutions.

cater

/ˈkeɪtə/
v. 满足需要; 迎合;
考虑 (*for, to*)

3. *System design with reuse*. During this phase, the framework of the system is designed or an existing framework is reused. The designers take into account the components that are reused and organize the framework to **cater** to this. Some new software may have to be designed if reusable components are not available.

procure

/prəʊkjuə/
v. 取得, 获得;
采办

4. *Development and integration*. Software that cannot be externally **procured** is developed, and the components and COTS systems are integrated to create the new system. System integration, in this model, may be part of the development process rather than a separate activity.

Component-based software engineering has the obvious advantage of reducing the amount of software to be developed and so reducing cost and risks. It usually also leads to faster delivery of the software. However, requirements compromises are inevitable and this may lead to a system that

¹ *Given the requirements specification*: 在有需求规格说明的情况下。本句中的 *given* 作介词用, 引起短语作状语, 意思相当于“在有…的情况下”, “如果有…”。

does not meet the real needs of users. Furthermore, some control over the system evolution is lost as new versions of the reusable components are not under the control of the organization using them.

Exercises

I. Fill in the blanks with the information given in the text:

1. New software may be developed from scratch or through the use of existing systems and _____ software or system components.
2. The three generic process models discussed in the text are often complementary rather than mutually _____, especially for large systems development.
3. The waterfall model is so named because of the _____ from one phase to another in the software development process.
4. The fundamental development activities of the waterfall model are _____ analysis and definition, system and software design, implementation and unit testing, _____ and system testing, and operation and maintenance.
5. The two fundamental types of evolutionary development are exploratory development and _____ prototyping.
6. The evolutionary approach is often more effective than the waterfall approach in producing systems that meet the _____ needs of customers, but it is difficult to establish a(n) _____ system architecture using this approach.
7. CBSE is a(n) _____ approach to software development, which relies on a large base of reusable software components and an integrating _____ for these components.
8. While CBSE can reduce the amount of _____ to be developed and the associated cost and risks, it cannot avoid requirements _____, which may lead to a system that does not meet the real needs of users.

II. Translate the following terms or phrases from English into Chinese and vice versa:

- | | |
|-------------------------------------|---------------|
| 1. system specification | 11. 系统设计范例 |
| 2. unit testing | 12. 需求分析与定义 |
| 3. software life cycle | 13. 探索式编程方法 |
| 4. system validation testing | 14. 系统文件编制 |
| 5. evolutionary development process | 15. 瀑布模型 |
| 6. linear model | 16. 系统集成 |
| 7. program unit | 17. 商用现成软件 |
| 8. throwaway prototype | 18. 基于组件的软件工程 |
| 9. text formatting | 19. 软件维护工具 |
| 10. system evolution | 20. 软件复用 |

III. Fill in each of the blanks with one of the words given in the following list, making changes if necessary:

<i>aspect</i>	<i>software</i>	<i>environmental</i>	<i>most</i>
<i>different</i>	<i>practice</i>	<i>maintenance</i>	<i>type</i>
<i>adapt</i>	<i>modify</i>	<i>way</i>	<i>response</i>
<i>expensive</i>	<i>fault</i>	<i>necessary</i>	<i>system</i>

There are three different types of software maintenance. Firstly, there is maintenance to repair software _____. Coding errors are usually relatively cheap to correct; design errors are more _____ as they may involve rewriting several program components. Requirements errors are the _____ expensive to repair because of the extensive system redesign that may be _____. Secondly, there is maintenance to adapt the software to a _____ operating environment. This type of maintenance is required when some _____ of the system's environment such as the hardware, the platform operating _____ or other support software changes. The application system must be _____ to adapt it to cope with these environmental changes. And thirdly, there is _____ to add to or modify the system's functionality. This _____ of maintenance is necessary when the system requirements change in _____ to organizational or business change. The scale of the changes required to the _____ is often much greater than for the other types of maintenance. In _____, there isn't a clear-cut distinction between these types of maintenance. When you _____ the system to a new environment, you may add functionality to

take advantage of new _____ features. Software faults are often exposed because users use the system in unanticipated _____. Changing the system to accommodate their way of working is the best way to fix these faults.

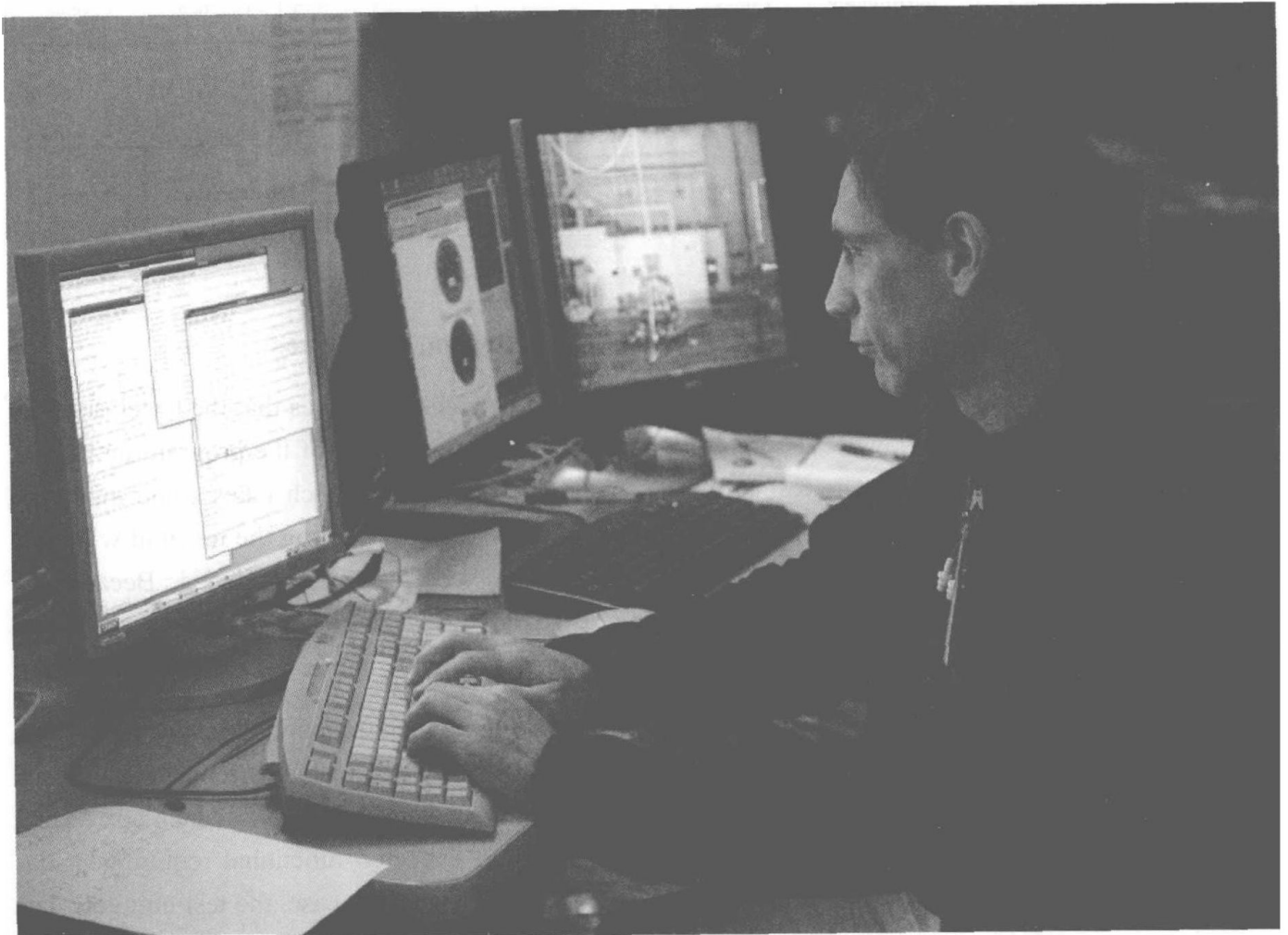
IV. Translate the following passage from English into Chinese:

Software processes are complex and, like all intellectual and creative processes, rely on people making decisions and judgments. Because of the need for judgment and creativity, attempts to automate software processes have met with limited success. Computer-aided software engineering (CASE) tools can support some process activities. However, there is no possibility, at least in the next few years, of more extensive automation where software takes over creative design from the engineers involved in the software process.

One reason the effectiveness of CASE tools is limited is because of the immense diversity of software processes. There is no ideal process, and many organizations have developed their own approach to software development. Processes have evolved to exploit the capabilities of the people in an organization and the specific characteristics of the systems that are being developed. For some systems, a very structured development process is required while for others a flexible, *agile* (敏捷的) process is likely to be more effective.

Section B

Software Testing Techniques



intermix

/,intə'miks/

v. (使) 混合 (*with*)

coverage /'kʌvərɪdʒ/

n. 覆盖 (范围)

exhaustive

/ɪg'zɔːstɪv/

a. 全面而彻底的,
详尽无遗的

execution path

执行路径

black box testing

黑盒测试 (法)

white box testing

白盒测试 (法)

There are many possible testing techniques, which can be **intermixed** to provide the best possible testing **coverage** and outcomes. No matter how extensive the testing is, it cannot be **exhaustive** and guarantee program/system correctness. It is never possible to test for all possible data inputs or code **execution paths**.

Testing techniques (strategies) can be classified according to various criteria. The following discussion is based on five criteria: (1) visibility, (2) automation, (3) partitioning, (4) coverage, and (5) scripting (Figure 5B-1). The criteria are not independent. For example, the two partitioning techniques apply mostly to **black box testing**, but not really to **white box**

regression

/ri'greʃən/

n. 回归

regression test

回归测试

exercising test

压力测试

equivalence

/i'kwiveləns/

n. 等价; 相等

equivalence**partitioning**

等价(类)划分

boundary value

[数] 边界值

testing. The scripting techniques (regression tests and exercising tests) are automatic tests.

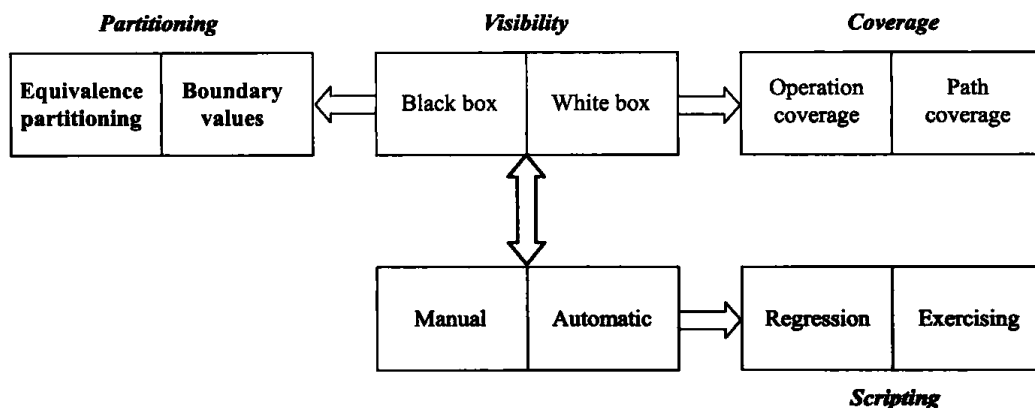


Figure 5B-1: Testing Techniques

spec /spek/

n. 〈口〉明确说明;

规格, 规范

(= specification)

Black box testing (or testing to specs¹) assumes that the tester does not know or chooses to ignore the internal workings of the program unit under test. The test unit is considered a black box, which takes some input and produces some output. The testing is done by feeding the test unit with data inputs and verifying that the expected output is produced. Because no implementation knowledge is required, black box testing can be conducted by the users. Accordingly, this is the main technique of acceptance testing.

The black box technique tests system functionality, or the system functional outputs, to be precise. It is also applicable for constraint testing, such as system performance or security. A special target of black box testing is missing functionality, i.e. when an expected functional requirement has not been implemented in the software. In such a case, the test unit gets data to trigger such functionality and fails because the code to do the job is not there.

White box testing (or testing to code) is the opposite of black box testing. The test unit is considered a white box or, better, a transparent glass box that reveals its content. In this approach, the tester studies the code and decides on data inputs that are able to “exercise” selected execution paths in the code. Black box testing must normally be followed by white box testing to **pinpoint** the errors discovered when testing to specs.

pinpoint /'pinpoint/

v. 精确地确定…的

位置; 确定

Unlike black box testing, white box testing is not restricted to code

¹ testing to specs: 按照规格测试。介词 to 表示“按”、“按照”。

artifact /'ɑ:tɪfækt/

n. 人工制品,
制造物

walkthrough

/'wɔ:kθru:/

n. 走查

counteract

/,kaunte'rækt/

v. 对...起反作用;
对抗; 抵消

partition /pɑ:'tɪʃən/

n. 分割; 划分; (分
割而成的) 部分

forfeit /'fɔ:fit/

v. 丧失, 失去

per se /,pe:'sei, -'si:/

(拉) 自身, 本身

integer /'ɪntɪdʒə/

n. 整数

testing. It is equally suitable for other development **artifacts**, such as design models and specification documents. Testing of this kind uses review techniques, such as **walkthroughs** and inspections. An interesting discovery possible with white box testing, but not with black box, is the existence of “dead code”—statements in the program that cannot be exercised or will never be used no matter what input is given to the code.

The equivalence partitioning and boundary value techniques apply mostly to black box testing, although white box testing can also benefit from them. They are specific approaches to the black box technique to **counteract** the impossibility of conducting exhaustive tests.

Equivalence partitioning groups data inputs (and, implicitly, data outputs) into **partitions** constituting homogeneous test targets. The assumption is that testing with any one member of the partition is as good as¹ testing with other members. Therefore, testing with the other members can be **forfeited**. Choosing the homogeneous partitions is the main difficulty. Partitions must be chosen with a good knowledge of data and the application demands for data. In this sense, partitioning **per se** is not really black box (it is not even a testing technique as such). Black box is the testing technique that equivalence partitioning supports.

Boundary value is merely an additional data analysis technique to assist in equivalence partitioning and, consequently, to assist in black box testing. Boundary values are extreme cases within equivalence partitions. For example, if the partition is a set of **integer** values from 1 to 100, the boundary value analysis will recommend tests to be done on the values on the edges of the partition, i.e. for -1, 0, and +1 as well as for 99, 100, and 101. Naturally, the expected outcome for testing on -1 and 101 would be an error condition.

Coverage techniques determine how much code is going to be exercised by a white box test. Operation coverage, called also method coverage, ensures that each operation in the code is exercised at least once by the white box test. Operation coverage is a modern object-oriented substitute for statement and branch coverage, which apply in procedural programming languages.

Path coverage aims at numbering possible execution paths in the program and exercising them one by one. Clearly, the number of such paths

¹ *as good as*: 与...几乎一样; 实际上等于。

indefinite

/in'definit/

a. 不确定的,
不定的

is **indefinite** in large programs. The tester is only able to define the most critical and most frequently used paths, and only these paths will be exercised.

test script

测试脚本

Testing can be manual or automatic. A human tester who interacts with the application under test conducts manual testing. The tester launches the application, systematically executes its functions, and observes the results. The execution steps are conducted according to a predefined **test script**. The test script defines step-by-step testing actions and expected outcomes. **Use cases** and other requirements specification documents are used to write test scripts.

use case

用例

baseline /'beislain/n. 基线; 基准;
基础

The main problem with manual tests is that, in most practical situations, they are performed on “live data.” There is no fixed **baseline** data predefined to guarantee the same output for repeating execution of manual tests. For this reason, expected output is not always defined precisely in the test scripts. Frequently, the output is not even presented to the screen, but it is **manifested** in database changes. This forces the tester to write and execute SQL scripts or other **test harnesses** to check the results of test actions.¹

manifest

/'mænifest/

v. 显示; 使显现

test harness

测试框架

Manual tests are expensive to prepare, execute, and manage. They are unable to satisfy the “volume” demands of testing. Automated testing employs software testing tools to execute large volumes of tests without human participation. The tools are also able to produce necessary post-test reports to facilitate management of test outcomes. Naturally the **preparatory** tasks, such as deciding what to test, programming some test scripts, and setting up the testing environment, must still be done by human experts.

preparatory

/pri'pæretəri; -tɔ:ri/

a. 准备性的, 预备性的

As indicated in Figure 5B-1, automated testing can be divided into regression testing and exercising testing. Regression testing is a popular term to mean repetitive execution of the same test scripts on the same baseline data to verify that previously accepted functionality of the system has not been broken by successive changes to the code, i.e. changes seemingly unrelated to the tested functionality.

Regression testing is performed by automatic execution of pre-recorded test scripts at scheduled test times. Original test scripts are

¹ This forces the tester to write and execute SQL scripts or other test harnesses to check the results of test actions.: 这迫使测试员编写并运行 SQL 脚本或其他测试框架, 来检查测试行动的结果。

recorded by a capture/playback tool in the process of capturing the human tester's actions on the application under test. Regression testing is a playback activity of playing back the scripts. In many cases, the tool-captured scripts are modified (improved) by a test programmer to allow regression testing of features, which the tool could not record automatically.

Exercising, for the lack of a better term, is really an automated coverage testing. A tool that implements exercising testing generates automatically and randomly various possible actions that otherwise could be performed by the user on the application under test. This could be compared to a mad user hitting any possible key on the keyboard, selecting any possible menu item, pressing any available command button, etc.

Generated actions are recorded (captured) in a script, sometimes called the best script, which is **deemed** to be able to exercise the application. Any errors or application failures are recorded as well and a separate script is generated for actions that led to the problem. This is called a defect script. The defect script can be played back at any time to reproduce the error and try to determine the reason behind it.

Because both regression and exercising tests use capture/playback tools, which result in programs in some scripting language, they can feed off each other, thus creating a powerful automated testing environment.¹ In practice, the difficulty with automated testing is not in conducting it, but in running **consecutive** tests on a stable testing environment, with an identical **testbed** database, with the same state of open and active applications on the test workstation, with predictable network speed, with network broadcasting to the workstation disabled, with the same desktop appearance (including video resolution, desktop colors, **fonts**), etc. Finally, all automatic tests should clean up after themselves and re-create the database and application testbed as it was before the tests started.

deem /'di:m/
v. 认为, 视为

consecutive
/kən'sekjutiv/
a. 连续的
testbed /'testbed/
n. 试验台, 测试台
font /fɒnt/
n. 字体

¹ *Because both regression and exercising tests use capture/playback tools, which result in programs in some scripting language, they can feed off each other, thus creating a powerful automated testing environment.*: 回归测试和压力测试都使用捕获/回放工具, 产生以某种脚本语言编成的程序, 因此, 它们可以互相利用, 这样就创建了一个功能强大的自动测试环境。feed off 的基本意思为“以…作为食物(或能量等)的来源”、“从…中得到滋养”。

Exercises

I. Fill in the blanks with the information given in the text:

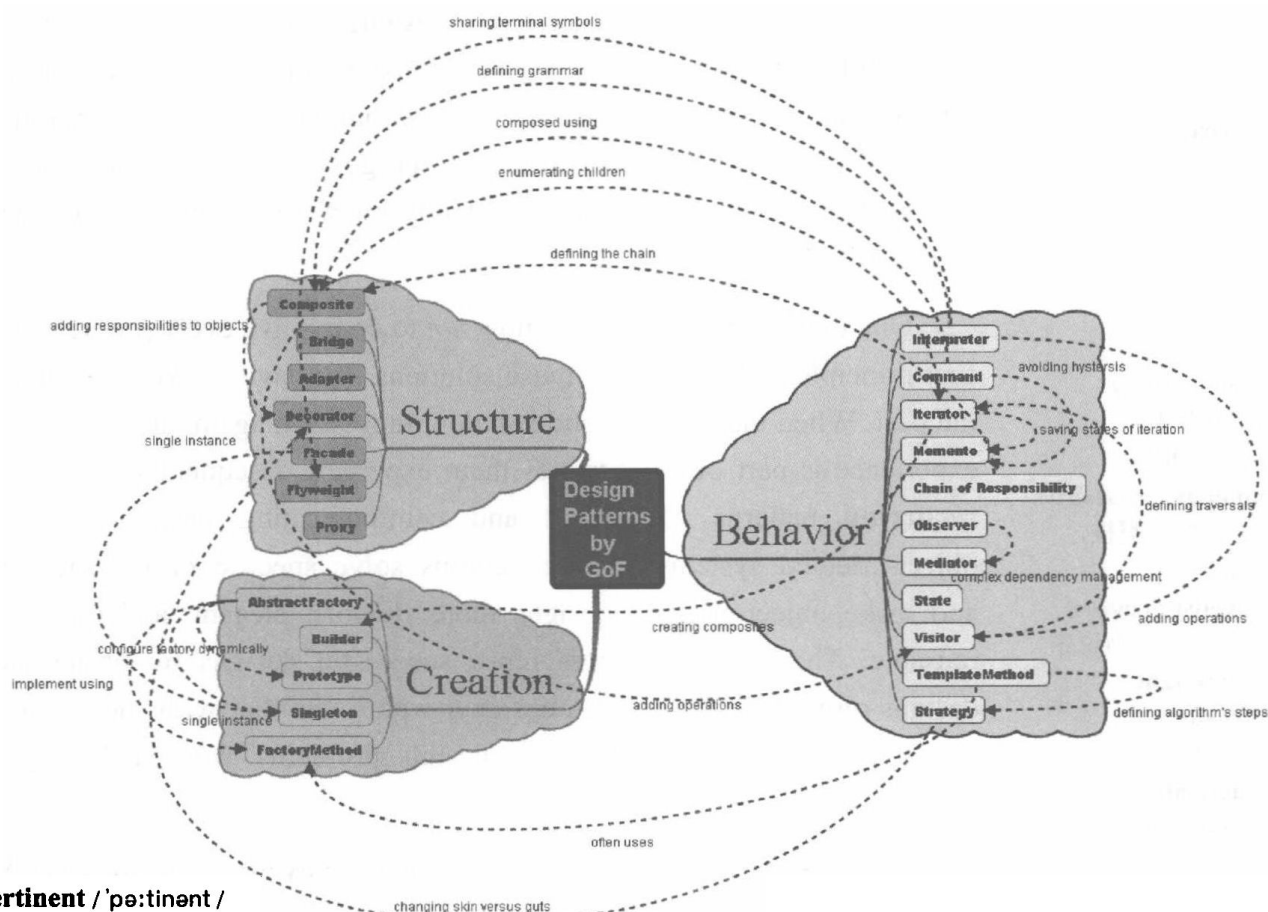
1. Black box testing is the main technique of _____ testing.
2. Black box testing is testing to specs, whereas white box testing is testing to _____.
3. Testing techniques can be classified according to various criteria, including visibility, _____, partitioning, _____, and scripting, on which the discussion in the text is based.
4. Automated testing can be divided into _____ testing and exercising testing.

II. Translate the following terms or phrases from English into Chinese and vice versa:

- | | |
|-----------------------------|--------------|
| 1. black box testing | 6. 捕获 / 回放工具 |
| 2. acceptance testing | 7. 视频分辨率 |
| 3. code execution path | 8. 白盒测试 |
| 4. test harness | 9. 测试脚本 |
| 5. equivalence partitioning | 10. 用例 |

Section C

What Is a Design Pattern¹



pertinent /'pɜ:tinent/

a. 相关的

factor /'fæktə/

v. 把...分解成 (into)

granularity

/,grænju'lærɪti/

n. (颗) 粒度

inheritance hierarchy

继承层次

Designing object-oriented software is hard, and designing *reusable* object-oriented software is even harder. You must find **pertinent** objects, **factor** them into classes at the right **granularity**, define class interfaces and **inheritance hierarchies**, and establish key relationships among them.² Your design should be specific to the problem at hand but also general

¹ “模式” (pattern) 的概念最早由建筑大师 Christopher Alexander 于 20 世纪 70 年代提出, 应用于建筑领域。20 世纪 80 年代中期, Ward Cunningham 和 Kent Beck 将其思想引入软件领域。软件业界的模式热最早源自 Erich Gamma、Richard Helm、Ralph Johnson 和 John Vlissides (在文献引用中常被称为 GoF, 即“四人帮”) 的经典著作 *Design Patterns: Elements of Reusable Object-Oriented Software* (本文摘自该书; 中译本《设计模式: 可复用面向对象软件的基础》已由机械工业出版社出版)。如今, 模式已成为软件工程领域内的一个热门话题, 在计算机领域的影响超过了在建筑界的影响。

² You must find pertinent objects, factor them into classes at the right granularity, define class interfaces and inheritance hierarchies, and establish key relationships among them.: 你必须找到相关的对象, 以适当的粒度将它们归类, 定义类的接口和继承层次, 并建立对象之间的关键关系。

enough to address future problems and requirements. You also want to avoid redesign, or at least minimize it. Experienced object-oriented designers will tell you that a reusable and flexible design is difficult if not impossible to get “right” the first time. Before a design is finished, they usually try to reuse it several times, modifying it each time.

Yet experienced object-oriented designers do make good designs. Meanwhile new designers are overwhelmed by the options available and tend to fall back on¹ non-object-oriented techniques they’ve used before. It takes a long time for novices to learn what good object-oriented design is all about. Experienced designers evidently know something inexperienced ones don’t. What is it?

One thing expert designers know *not* to do is solve every problem from first principles.² Rather, they reuse solutions that have worked for them in the past. When they find a good solution, they use it again and again. Such experience is part of what makes them experts. Consequently, you’ll find **recurring** patterns of classes and communicating objects in many object-oriented systems.³ These patterns solve specific design problems and make object-oriented designs more flexible, elegant, and ultimately reusable. They help designers reuse successful designs by basing new designs on prior experience. A designer who is familiar with such patterns can apply them immediately to design problems without having to rediscover them.

An **analogy** will help illustrate the point. **Novelists** and **playwrights** rarely design their plots from scratch. Instead, they follow patterns like “**Tragically Flawed Hero**”⁴ (*Macbeth*⁵, *Hamlet*⁶, etc.) or “The Romantic Novel” (countless **romance** novels). In the same way, object-oriented designers follow patterns like “represent states with objects” and “decorate objects so you can easily add/remove features.” Once you know the pattern, a lot of design decisions follow automatically.

novice /ˈnɒvɪs/
n. 新手, 初学者

recur /rɪˈkeː/
v. 再发生;
反复出现
analogy /əˈnælədʒi/
n. 比拟, 类推,
类比

novelist /ˈnɒvəlɪst/
n. (长篇) 小说家

playwright
/ˈpleɪraɪt/
n. 剧作家

tragic(al)
/ˈtrædʒɪk(əl)/
a. 悲剧(性)的;
悲惨的

flawed /flɔːd/
a. 有缺点的, 有瑕疵的

romance
/ˈrəʊmæns, ˈrɔːmæns/
n. 浪漫文学, 传奇文学

¹ *fall back on*: 借助于; 依赖, 依靠。

² *One thing expert designers know not to do is solve every problem from first principles.*: 内行的设计者知道不要做的一件事是: 解决任何问题都从头做起。

³ *Consequently, you’ll find recurring patterns of classes and communicating objects in many object-oriented systems.*: 因此, 你会在许多面向对象系统中看到类和相互通信的对象的重复模式。

⁴ *Tragically Flawed Hero*: 带有悲剧性缺陷的主人公(指悲剧主人公存在导致自身毁灭的性格上的缺陷, 如骄傲、猜疑等)。

⁵ *Macbeth*: 麦克白(亦译作麦克佩斯), 英国伟大的剧作家、诗人莎士比亚(1564—1616)的同名悲剧中的主人公, 其悲剧在于野心战胜了善良的天性。

⁶ *Hamlet*: 哈姆雷特(亦译作汉姆雷特), 莎士比亚的同名悲剧中的主人公, 属于性格优柔寡断的思考型人物。

déjà vu /,deɪʒɑːˈvjuː/
〈法〉似曾经历的
错觉

We all know the value of design experience. How many times have you had design **déjà vu**—that feeling that you've solved a problem before but not knowing exactly where or how? If you could remember the details of the previous problem and how you solved it, then you could reuse the experience instead of rediscovering it. However, we don't do a good job of recording experience in software design for others to use.

The purpose of this book is to record experience in designing object-oriented software as design patterns. Each design pattern systematically names, explains, and evaluates an important and recurring design in object-oriented systems. Our goal is to capture design experience in a form that people can use effectively. To this end we have documented some of the most important design patterns and present them as a catalog.¹

compromise
/'kɒmprəmaɪz/
v. 连累; 危及,
损害
intent /in'tent/
n. 意图, 目的

Design patterns make it easier to reuse successful designs and architectures. Expressing proven techniques as design patterns makes them more accessible to developers of new systems. Design patterns help you choose design alternatives that make a system reusable and avoid alternatives that **compromise** reusability. Design patterns can even improve the documentation and maintenance of existing systems by furnishing an explicit specification of class and object interactions and their underlying **intent**. Put simply, design patterns help a designer get a design “right” faster.

handle /'hændl/
n. 句柄; (文件、对
象等的) (名) 称
(编) 号, 名 (字
编) 号

Christopher Alexander² says, “Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice”. Even though Alexander was talking about patterns in buildings and towns, what he says is true about object-oriented design patterns. Our solutions are expressed in terms of objects and interfaces instead of walls and doors, but at the core of both kinds of patterns is a solution to a problem in a context.

In general, a pattern has four essential elements:

1. The pattern name is a **handle** we can use to describe a design

¹ To this end we have documented some of the most important design patterns and present them as a catalog.: 为此目的, 我们编写了一些最重要的设计模式, 并以编目分类的形式将它们展现出来。

² Christopher Alexander: 克里斯托弗·亚历山大(1936—), 出生于奥地利首都维也纳, 先后毕业于剑桥大学和哈佛大学, 获建筑学博士学位。他于20世纪70年代提出“模式”(pattern)的概念, 应用于建筑领域, 在建筑学界引起强烈反响。但是, 他(以及他的著作)对计算机科学的影响甚至远远超过对建筑学的影响。他的“模式语言”理论被应用在面向对象的程序设计上, 对软件设计起到了极其深远的影响。

trade-off /ˈtreɪdɒf/
n. 平衡, 权衡; (公平) 交易

symptomatic
/ˌsɪmptəˈmætɪk/
a. 作为征候 (或征兆) 的; 表明的 (*of*)

template /ˈtemplɪt/
n. 模板

portability
/ˌpɔːteˈbɪlɪti/
n. 可移植性; 便于携带

problem, its solutions, and consequences in a word or two. Naming a pattern immediately increases our design vocabulary. It lets us design at a higher level of abstraction. Having a vocabulary for patterns lets us talk about them with our colleagues, in our documentation, and even to ourselves. It makes it easier to think about designs and to communicate them and their **trade-offs** to others. Finding good names has been one of the hardest parts of developing our catalog.

2. The problem describes when to apply the pattern. It explains the problem and its context. It might describe specific design problems such as how to represent algorithms as objects. It might describe class or object structures that are **symptomatic** of an inflexible design. Sometimes the problem will include a list of conditions that must be met before it makes sense to apply the pattern.
3. The solution describes the elements that make up the design, their relationships, responsibilities, and collaborations. The solution doesn't describe a particular concrete design or implementation, because a pattern is like a **template** that can be applied in many different situations. Instead, the pattern provides an abstract description of a design problem and how a general arrangement of elements (classes and objects in our case) solves it.
4. The consequences are the results and trade-offs of applying the pattern. Though consequences are often unvoiced when we describe design decisions, they are critical for evaluating design alternatives and for understanding the costs and benefits of applying the pattern. The consequences for software often concern space and time trade-offs. They may address language and implementation issues as well. Since reuse is often a factor in object-oriented design, the consequences of a pattern include its impact on a system's flexibility, extensibility, or **portability**. Listing these consequences explicitly helps you understand and evaluate them.

Point of view affects one's interpretation of what is and isn't a pattern. One person's pattern can be another person's primitive building block. For this book we have concentrated on patterns at a certain level of abstraction. The design patterns in this book are *descriptions of communicating objects and classes that are customized to solve a general design problem in a particular context*.

abstract

/æb'strækt/

v. 把...抽象出来;
提出, 抽出

A design pattern names, **abstracts**, and identifies the key aspects of a common design structure that make it useful for creating a reusable object-oriented design. The design pattern identifies the participating classes and instances, their roles and collaborations, and the distribution of responsibilities. Each design pattern focuses on a particular object-oriented design problem or issue. It describes when it applies, whether it can be applied in view of other design constraints, and the consequences and trade-offs of its use. Since we must eventually implement our designs, a design pattern also provides sample C++ and (sometimes) Smalltalk code to illustrate an implementation.

mainstream

/'meɪnstri:m/

a. 主流的

Although design patterns describe object-oriented designs, they are based on practical solutions that have been implemented in **mainstream** object-oriented programming languages like Smalltalk and C++ rather than procedural languages (Pascal, C, Ada) or more dynamic object-oriented languages (CLOS¹, Dylan², Self³). We chose Smalltalk and C++ for **pragmatic** reasons: Our day-to-day experience has been in these languages, and they are increasingly popular.

pragmatic

/præɡ'mætɪk/

a. 讲究实际的,
重实效的;
实用主义的

The choice of programming language is important because it influences one's point of view. Our patterns assume Smalltalk/C++-level language features, and that choice determines what can and cannot be implemented easily. If we assumed procedural languages, we might have included design patterns called "Inheritance," "Encapsulation," and "Polymorphism." Similarly, some of our patterns are supported directly by the less common object-oriented languages. CLOS has multi-methods, for example, which lessen the need for a pattern such as Visitor. In fact, there are enough differences between Smalltalk and C++ to mean that some patterns can be expressed more easily in one language than the other.

Exercises

I. Fill in the blanks with the information given in the text:

1. A design pattern generally has four essential elements: the pattern

¹ CLOS: 公共 Lisp 对象系统 (Common Lisp Object System 的首字母缩略)。

² Dylan: 1992 年由苹果公司发布的一种面向对象语言。

³ Self: 一种基于原型的面向对象程序设计语言, 于 1986 年由施乐帕洛阿尔托研究中心的戴维·昂加尔 (David Ungar) 和兰迪·史密斯 (Randy Smith) 给出了最初的设计。

name, _____, solution, and consequences.

2. Reusability is often a factor in object-oriented design, so the _____ of a pattern include its impact on a system's flexibility, extensibility, or portability.

3. The solution of a design pattern describes the elements that make up the _____, their relationships, responsibilities, and collaborations.

4. With reusable object-oriented software, your design should be _____ to the problem at hand but also _____ enough to address future problems and requirements.

II. Translate the following terms or phrases from English into Chinese and vice versa:

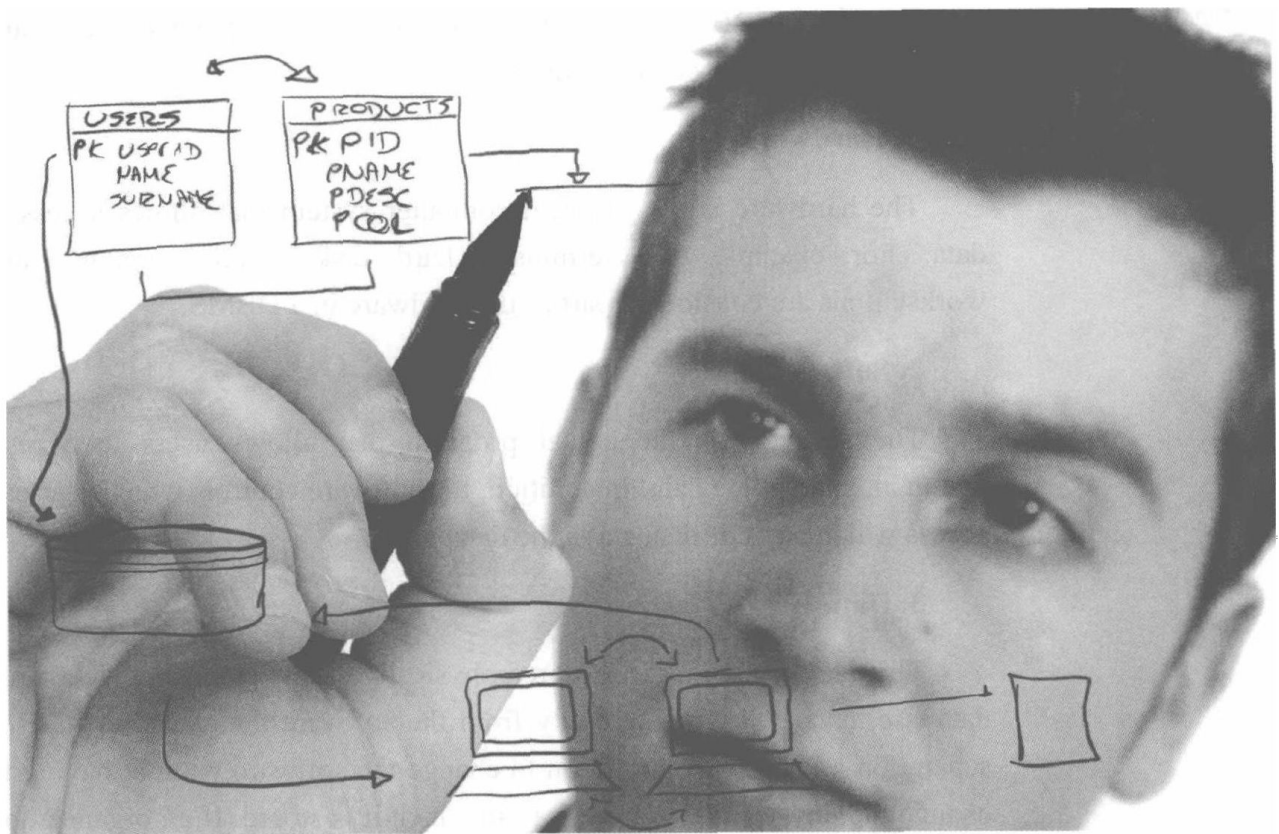
- | | |
|---------------------------------|------------|
| 1. procedural language | 5. 设计模式 |
| 2. common design structure | 6. 可复用软件 |
| 3. class and object interaction | 7. 面向对象的系统 |
| 4. design constraint | 8. 继承层次 |

Unit 6 Database

(数据库)

Section A

Database Overview



I. Introduction

flat file

平面文件,
展开文件

payroll / 'peɪrəʊl /

n. 工资表; 在职人员名单; 工薪总额

Data storage traditionally used individual, unrelated files, sometimes called **flat files**. In the past, each application program in an organization used its own file. In a university, for example, each department might have its own set of files: the record office kept a file about the student information and their grades, the financial aid office kept its own file about students that needed financial aid to continue their education, the scheduling office kept the names of the professors and the courses they were teaching, the **payroll** department kept its own file about the whole

staff (including professors), and so on. Today, however, all of these flat files can be combined in a single entity, the database for the whole university.

Although it is difficult to give a universally agreed definition of a database, we use the following common definition: a database is a collection of related, logically coherent, data used by the application programs in an organization.

II. Database Management Systems

A database management system (DBMS) defines, creates, and maintains a database. The DBMS also allows controlled access to data in the database. A DBMS is a combination of five components: hardware, software, data, users, and procedures.

1. Hardware

The hardware is the physical computer system that allows access to data. For example, the terminals, hard disk, main computer, and workstations are considered part of the hardware in a DBMS.

2. Software

The software is the actual program that allows users to access, maintain, and update data. In addition, the software controls which user can access which parts of the data in the database.

3. Data

The data in a database is stored physically on the storage devices. In a database, data is a separate entity from the software that accesses it. This separation allows an organization to change the software without having to change the physical data or the way in which it is stored. If an organization decides to use a DBMS, then all the information needed by the organization should be kept together as one entity, to be accessible by the software in the DBMS.

4. Users

The term users in a DBMS has a broad meaning. We can divide users into two categories: end users and application programs.

End users are those humans who can access the database directly to get information. There are two types of end users: database administrators (DBAs) and normal users. Database administrators have the maximum level

coherent

/kəʊ'hɪərənt/

a. 相干的; 一致的;
协调的

database

management

system

数据库管理系统

end user

最终用户,

终端用户

administrator

/əd'mɪnɪstreɪtə/

n. (系统、程序
等的) 管理员

revoke /ri'vəuk/
v. 撤销; 废除

of privileges and can control other users and their access to the DBMS, grant some of their privileges to somebody else, but retain the ability to **revoke** them at any time. A normal user, on the other hand, can only use part of the database and has limited access.

paycheck /'peɪtʃek/
n. 〈美〉支付薪金
(或工资的)支票

The other users of data in a database are application programs. Applications need to access and process data. For example, a payroll application program needs to access part of the data in a database to create **paychecks** at the end of the month.

5. Procedures

The last component of a DBMS is a set of procedures or rules that should be clearly defined and followed by the users of the database.

database architecture
数据库体系结构
conceptual
/kən'septʃuəl/
a. 概念的

III. Database Architecture

The American National Standards Institute Standards Planning and Requirements Committee (ANSI/SPARC¹) has established a three-level architecture for a DBMS: internal, **conceptual**, and external (Figure 6A-1).

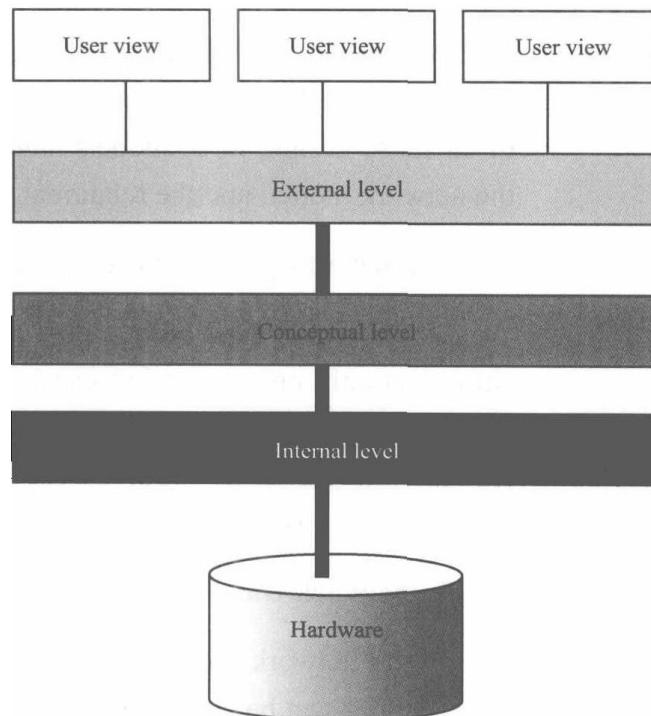


Figure 6A-1: Database Architecture

¹ **ANSI/SPARC**: 美国国家标准协会标准计划与需求委员会 (American National Standards Institute Standards Planning and Requirements Committee 的首字母缩略)。

1. Internal Level

The internal level determines where data is actually stored on the storage devices. This level deals with low-level access methods and how bytes are transferred to and from storage devices. In other words, the internal level interacts directly with the hardware.

2. Conceptual Level

The conceptual level defines the logical view of the data. The data model is defined on this level, and the main functions of the DBMS, such as queries, are also on this level. The DBMS changes the internal view of data to the external view that users need to see. The conceptual level is an **intermediary** and frees users from dealing with the internal level.

intermediary

/,intə'mi:diəri/

n. 媒介(物); 调解人; 中间人

3. External Level

The external level interacts directly with the user (end users or application programs). It changes the data coming from the conceptual level to a format and view that is familiar to the users.

IV. Database Models

A database model defines the logical design of data. The model also describes the relationships between different parts of the data. In the history of database design, three models have been in use: the hierarchical model, the network model, and the relational model.

1. Hierarchical Database Model

In the hierarchical model, data is organized as an **inverted** tree. Each entity has only one parent but can have several children. At the top of the hierarchy, there is one entity, which is called the root. Figure 6A-2 shows a logical view of an example of the hierarchical model. The hierarchical model is now **obsolete**.

inverted /in've:tɪd/

a. 反向的; 倒置的

obsolete

/ˈɒbsəli:t, ˌɒbsə'li:t/

a. 废弃的; 淘汰的; 过时的

2. Network Database Model

In the network model, the entities are organized in a graph, in which some entities can be accessed through several paths (Figure 6A-3). There is no hierarchy. This model is now also obsolete.

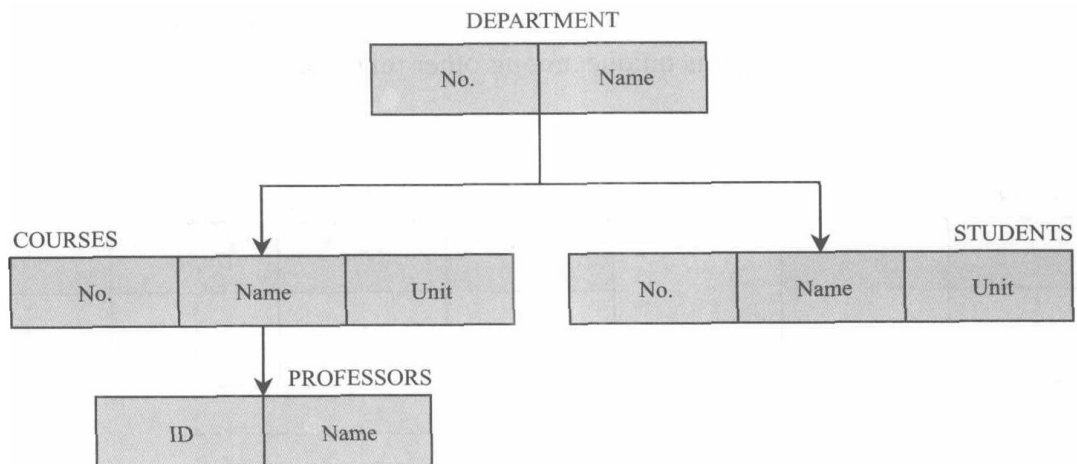


Figure 6A-2: An Example of the Hierarchical Model Representing a University

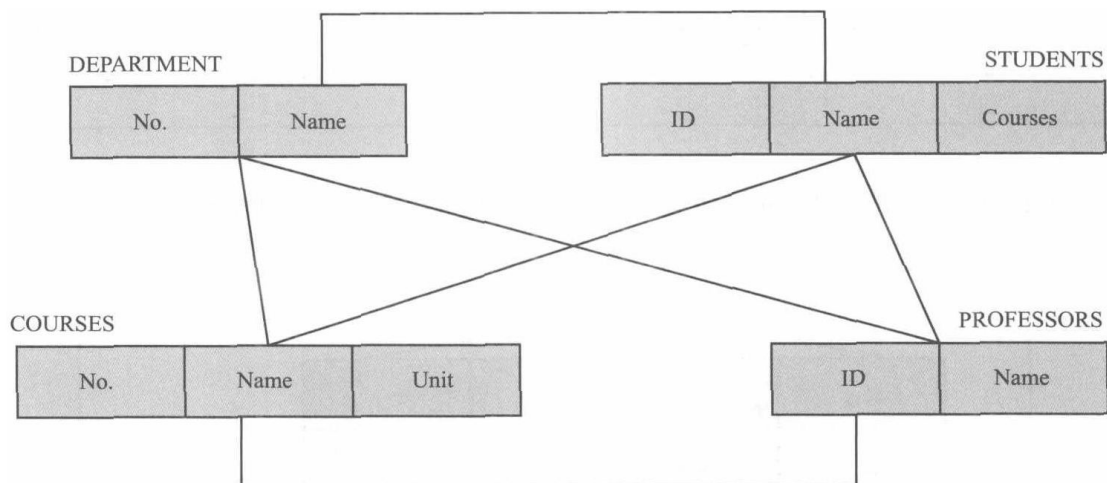


Figure 6A-3: An Example of the Network Model Representing a University

3. Relational Database Model

two-dimensional table 二维表

relational database management system 关系数据库管理系统

In the relational model, data is organized in **two-dimensional tables** called relations. There is no hierarchical or network structure imposed on the data. The tables or relations are, however, related to each other (Figure 6A-4). The **relational database management system (RDBMS)** organizes the data so that its external view is a set of relations or tables. This does not mean that data is stored as tables: the physical storage of the data is independent of the way in which the data is logically organized. Figure 6A-5 shows an example of a relation. A relation in an RDBMS has the

following features:

- **Name.** Each relation in a relational database should have a name that is unique among other relations.

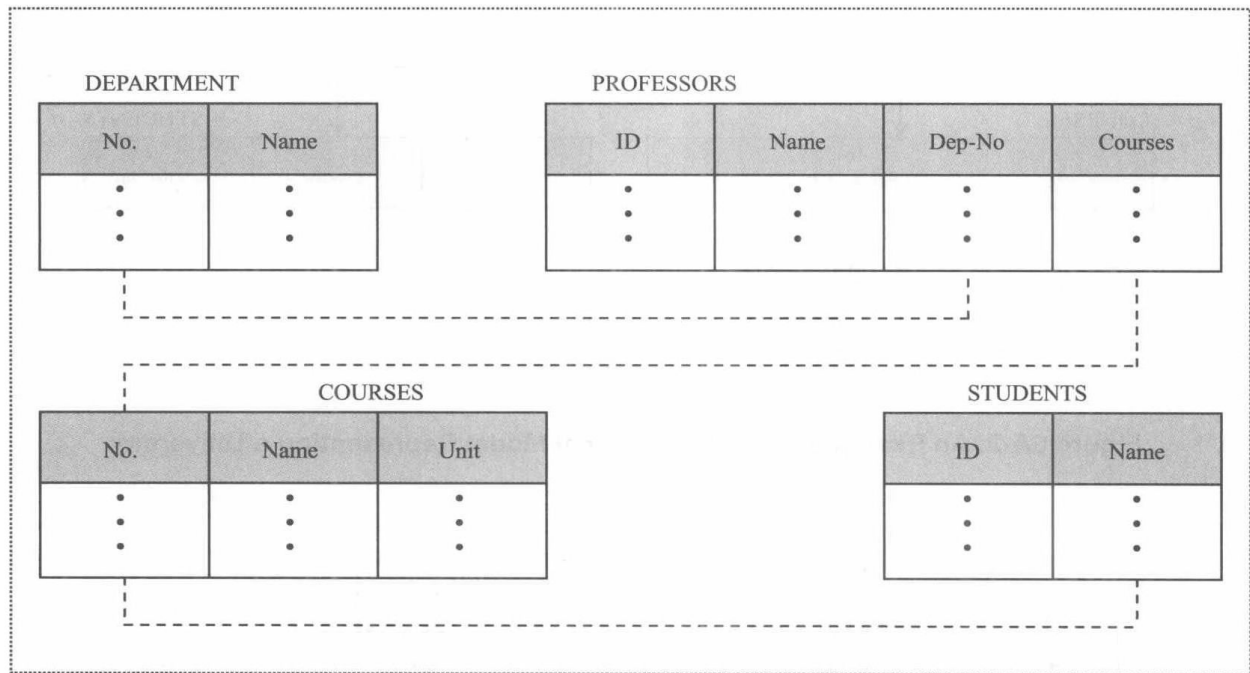


Figure 6A-4: An Example of the Relational Model Representing a University

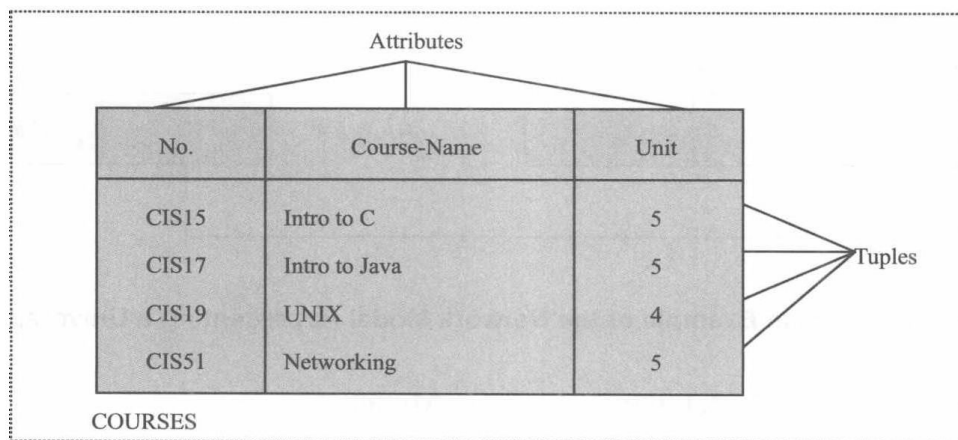


Figure 6A-5: An Example of a Relation

- **Attributes.** Each column in a relation is called an attribute. The attributes are the column headings in the table in Figure 6A-5. Each attribute gives meaning to the data stored under it. Each column in the table must have a name that is unique in the scope of the relation. The total number of attributes for a relation is called the degree of the relation. For example, in Figure 6A-5, the relation

tuple /'tʌpl, 'tju:pl/
n. 元组, 字节组

cardinality
/,kɑ:di'næliti/
n. 基数性; (集的)
势, (集的) 基数

wide area network
广域网

replicate /'replikeit/
v. 复制, 重复

pharmaceutic(al)
/,fɑ:me'sju:tik(əl)/
a. 制药的; 药(物)
的; 药用的

replica /'replike/
n. 复制品, 拷贝

intersection
/,inta'sekʃən/
n. 交, 相交, 交集;
交点

has a degree of 3. Note that the attribute names are not stored in the database: the conceptual level uses the attributes to give meaning to each column.

- **Tuples.** Each row in a relation is called a tuple. A tuple defines a collection of attribute values. The total number of rows in a relation is called the **cardinality** of the relation. Note that the cardinality of a relation changes when tuples are added or deleted. This makes the database dynamic.

The relational model is one of the common models in use today. The other two common models that are derived from the relational model are the distributed model and the object-oriented model.

4. Distributed Database Model

The distributed database model is not a new model, but is based on the relational model. However, the data is stored on several computers that communicate through the Internet or a private **wide area network**. Each computer (or site) maintains either part of the database or the whole database. In other words, data is either fragmented, with each fragment stored at one site, or data is **replicated** at each site.

In a fragmented distributed database, data is localized—locally used data is stored at the corresponding site. However, this does not mean that a site cannot access data stored at another site, but access is mostly local, but occasionally global. Although each site has complete control over its local data, there is global control through the Internet or a wide area network.

For example, a **pharmaceutical** company may have multiple sites in many countries. Each site has a database with information about its own employees, but a central personnel department could have control of all the databases.

In a replicated distributed database, each site holds an exact **replica** of another site. Any modification to data stored in one site is repeated exactly at every site. The reason for having such a database is security. If the system at one site fails, users at the site can access data at another site.

5. Object-Oriented Database Model

The relational database has a specific view of data that is based on the nature of the database's tuples and attributes. The smallest unit of data in a relational database is the **intersection** of a tuple and an attribute. However,

field / fi:ld /

n. 字段; 域; 信息组

some applications need to look at data as other forms, for example, to see data as a structure, such as a record composed of **fields**.

An object-oriented database tries to keep the advantages of the relational model and at the same time allows applications to access structured data. In an object-oriented database, objects and their relations are defined. In addition, each object can have attributes that can be expressed as fields.

social security number

社会安全(或保险、保障)号(码)

For example, in an organization, one could define object types for employee, department, and customer. The employee class could define the attributes of an employee object (first name, last name, **social security number**, salary, and so on) and how they can be accessed. The department object could define the attributes of the department and how they can be accessed. In addition, the database could create a relation between an employee object and a department object to **denote** that the employee works in that department.

denote / di'neut /

v. 表示; 意思是

V. Database Design

involved / in'vɒlvd /

a. (由于复杂或混乱而) 难处理的, 棘手的; 复杂的

The design of any database is a lengthy and **involved** task that can only be done through a step-by-step process. The first step normally involves a lot of interviewing of potential users of the database to collect the information needed to be stored and the access requirement of each department. The second step is to build an **entity-relationship model** (ERM) that defines the entities for which some information must be maintained, the attributes of these entities, and the relationship between these entities.

entity-relationship model

实体关系模型

The next step in design is based on the type of database to be used. In a relational database, the next step is to build relations based on the ERM and **normalize** the relations. Normalization is the process by which a given set of relations are transformed to a new set of relations with a more solid structure. Normalization is needed to allow any relation in the database to be represented, to allow a language like SQL to use powerful retrieval operations composed of atomic operations,¹ to remove **anomalies** in insertion, deletion, and updating, and to reduce the need for restructuring the database as new data types are added.

normalize

/'nɔ:məlaɪz /

v. 使正常(或标准、规范、正规、规格)化

anomaly / ə'nɒməli /

n. 异常(现象); 不按常规

¹ to use powerful retrieval operations composed of atomic operations: 使用由多个原子操作组成的功能强大的检索操作。atomic operations 意思是“原子操作”, 指不可分割的最小操作。

Exercises

I. Fill in the blanks with the information given in the text:

1. Data storage traditionally used individual, unrelated files, sometimes called _____ files.
2. According to the text, a DBMS is a combination of five components: hardware, software, _____, users, and procedures.
3. As far as a DBMS is concerned, users can refer to both end users and _____ programs, and end users can refer to both database _____ and normal users.
4. The three-level architecture for a DBMS discussed in the text consists of the internal, _____, and external levels.
5. In the relational database model, data is organized in two-dimensional _____ called relations.
6. A distributed database can be classified either as a _____ distributed database or as a _____ distributed database.
7. While trying to keep the advantages of the relational model, an object-oriented database allows applications to access _____ data.
8. An important step in the design of a database involves the building of a(n) _____ model that defines the entities for which some information must be maintained, the _____ of these entities, and the relationship between these entities.

II. Translate the following terms or phrases from English into Chinese and vice versa:

- | | |
|-------------------------------|---------------|
| 1. end user | 11. 二维表 |
| 2. atomic operation | 12. 数据属性 |
| 3. database administrator | 13. 数据库对象 |
| 4. relational database model | 14. 存储设备 |
| 5. local data | 15. 数据类型 |
| 6. object-oriented database | 16. 数据插入与删除 |
| 7. database management system | 17. 层次数据库模型 |
| 8. entity-relationship model | 18. 数据库体系结构 |
| 9. distributed database | 19. 关系数据库管理系统 |
| 10. flat file | 20. 全局控制总线 |

III. Fill in each of the blanks with one of the words given in the following list, making changes if necessary:

<i>equipment</i>	<i>important</i>	<i>access</i>	<i>center</i>
<i>available</i>	<i>text</i>	<i>use</i>	<i>every</i>
<i>develop</i>	<i>database</i>	<i>experimental</i>	<i>link</i>
<i>nonprofit</i>	<i>online</i>	<i>public</i>	<i>limited</i>

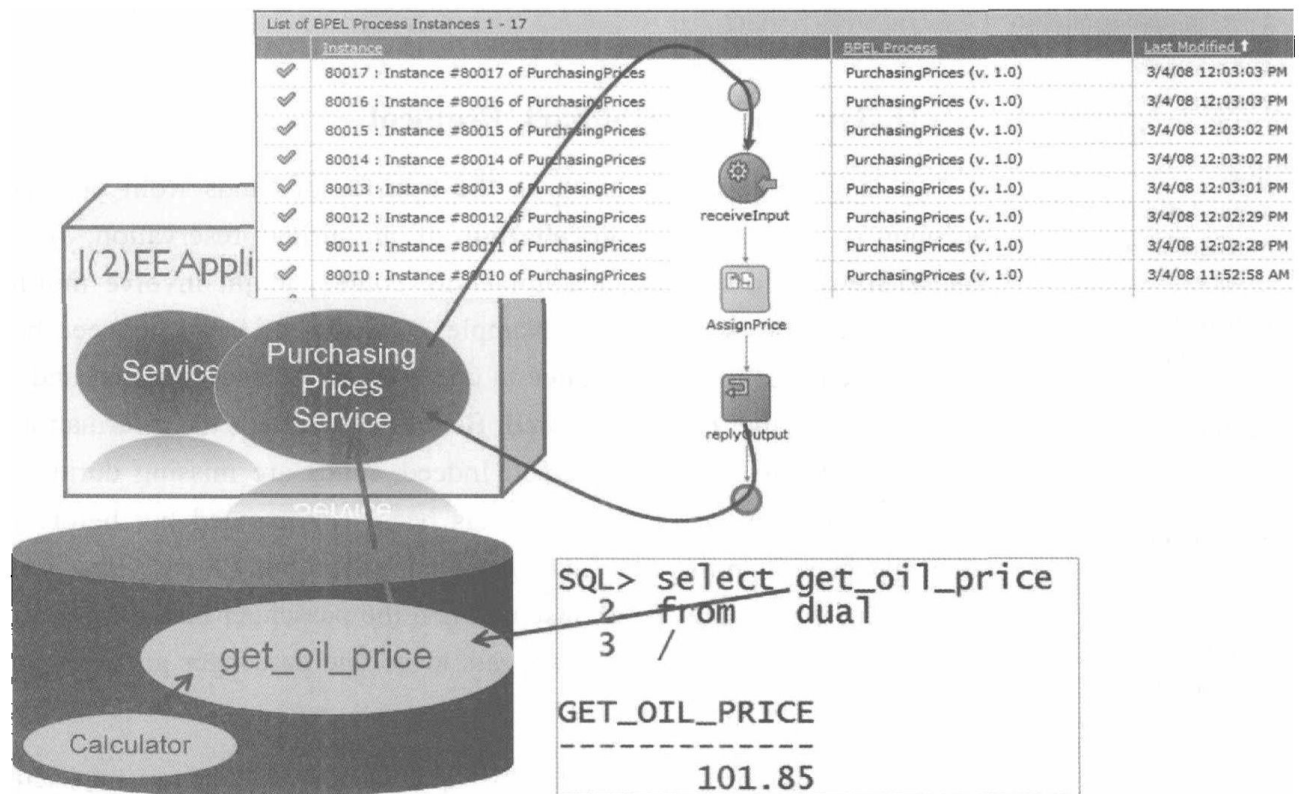
A database is any collection of data organized for storage in a computer memory and designed for easy _____ by authorized users. The data may be in the form of _____, numbers, or encoded graphics. Small databases were first _____ or funded by the U.S. government for agency or professional _____. In the 1960s, some databases became commercially _____, but their use was *funnelled* (传送) through a few so-called research _____ that collected information inquiries and handled them in *batches* (批, 批量). _____ databases—that is, databases available to anyone who could _____ up to them by computer—first appeared in the 1970s. Since their first, _____ appearance in the 1950s, databases have become so _____ that they can be found in almost _____ field of information. Government, military, and industrial _____ are often highly restricted, and professional databases are usually of _____ interest. A wide range of commercial, governmental, and _____ databases are available to the general _____, however, and may be used by anyone who owns or has access to the _____ that they require.

IV. Translate the following passage from English into Chinese:

In a relational database, the rows of a table represent records (collections of information about separate items) and the columns represent fields (particular attributes of a record). In conducting searches, a relational database matches information from a field in one table with information in a corresponding field of another table to produce a third table that combines requested data from both tables. For example, if one table contains the fields EMPLOYEE-ID, LAST-NAME, FIRST-NAME, and HIRE-DATE, and another table contains the fields DEPT, EMPLOYEE-ID, and SALARY, a relational database can match the EMPLOYEE-ID fields in the two tables to find such information as the names of all employees earning a certain salary or the departments of all employees hired after a certain date. In other words, a relational database uses matching values in two tables to relate information in one to information in the other. Microcomputer database products typically are relational databases.

Section B

Maintaining Database Integrity



I. Introduction

Inexpensive database management systems for personal use are relatively simple systems. They tend to have a single objective—to shield the user from the technical details of the database implementation. The databases maintained by these systems are relatively small and generally contain information whose loss or **corruption** would be inconvenient rather than **disastrous**. When a problem does arise, the user can usually correct the **erroneous** items directly or reload the database from a **backup** copy and manually make the modifications required to bring that copy up to date. This process might be inconvenient, but the cost of avoiding the inconvenience tends to be greater than the inconvenience itself. In any case, the inconvenience is restricted to only a few people, and any financial loss is generally limited.

In the case of large, multiuser, commercial database systems, however,

corruption

/kə'ɹʌpʃən/

n. 破坏; 腐化;
讹误

disastrous

/di'zɑ:stres/

a. 灾难性的

erroneous

/i'reʊniəs/

a. 错误的,
不正确的

backup

/'bækʌp/

a. 备份的, 后备的

devastating

/'devəsteɪtɪŋ/

a. 破坏性极大的, 毁灭性的

inadvertent

/ˌɪnəd'veɪtənt/

a. 因疏忽造成的; 粗心大意的

commit /kə'mɪt/

v. 提交, 委托; 承诺

cancellation

/ˌkænsə'leɪʃən/

n. 取消; 删去

registration

/ˌredʒɪ'streɪʃən/

n. 登记; 注册

balance /'bæləns/

n. 余额; 差额; 结算

decrement

/'dekɪmənt/

v. 减少, 减缩

increment

/'ɪnkɪmənt/

v. 增加, 增长

snapshot /'snæpʃɒt/

n. 快照; 瞬象, 瞬态图; 抽点打印

log /lɒg/

n. (运行) 记录; (系统) 日志

the stakes are much higher. The cost of incorrect or lost data can be enormous and can have **devastating** consequences. In these environments, a major role of the DBMS is to maintain the database's integrity by guarding against problems such as operations that for some reason are only partially completed or different operations that might interact **inadvertently** to cause inaccurate information in the database. It is this role of a DBMS that we address in this section.

II. The Commit/Rollback Protocol

A single transaction, such as the transfer of funds from one bank account to another, the **cancellation** of an airline reservation, or the **registration** of a student in a university course, might involve multiple steps at the database level. For example, a transfer of funds between bank accounts requires that the **balance** in one account be **decremented** and the balance in the other be **incremented**. Between such steps the information in the database might be inconsistent. Indeed, funds are missing during the brief period after the first account has been decremented but before the other has been incremented. Likewise, when reassigning a passenger's seat on a flight, there might be an instant when the passenger has no seat or an instant when the passenger list appears to be one passenger greater than it actually is.

In the case of large databases that are subject to heavy transaction loads, it is highly likely that a random **snapshot** will find the database in the middle of some transaction.¹ A request for the execution of a transaction or an equipment malfunction will therefore likely occur at a time when the database is in an inconsistent state.

Let us first consider the problem of a malfunction. The goal of the DBMS is to ensure that such a problem will not freeze the database in an inconsistent state. This is often accomplished by maintaining a **log** containing a record of each transaction's activities in a nonvolatile storage system, such as a magnetic disk. Before a transaction is allowed to alter the database, the alteration to be performed is first recorded in the log. Thus the log contains a permanent record of each transaction's actions.

The point at which all the steps in a transaction have been recorded in

¹ In the case of large databases that are subject to heavy transaction loads, it is highly likely that a random snapshot will find the database in the middle of some transaction.: 在事务量很大的大型数据库的情况下, 数据库在任意一个瞬间都极可能有某个事务正在处理当中。

commit point

提交点

the log is called the **commit point**. It is at this point that the DBMS has the information it needs to reconstruct the transaction on its own if that should become necessary. At this point the DBMS becomes committed to the transaction in the sense that it accepts the responsibility of guaranteeing that the transaction's activities will be reflected in the database.¹ In the case of an equipment malfunction, the DBMS can use the information in its log to reconstruct the transactions that have been completed (committed) since the last backup was made.

If problems should arise before a transaction has reached its commit point,² the DBMS might find itself with a partially executed transaction that cannot be completed. In this case the log can be used to roll back (undo) the activities actually performed by the transaction.³ In the case of a malfunction, for instance, the DBMS could recover by rolling back those transactions that were incomplete (noncommitted) at the time of the malfunction.

Rollbacks of transactions are not restricted, however, to the process of recovering from equipment malfunctions. They are often a part of a DBMS's normal operation. For example, a transaction might be terminated before it has completed all its steps because of an attempt to access **privileged** information, or it might be involved in a **deadlock** in which competing transactions find themselves waiting for data being used by each other. In these cases, the DBMS can use the log to roll back a transaction and thus avoid an erroneous database due to incomplete transactions.

privileged

/'privilidʒd/

a. 享有特权的;
特许的**deadlock** /'dedlɒk/

n. 死锁; 僵局

lurk /lɜ:k/

v. 潜伏; 暗藏; 潜行

To emphasize the delicate nature of DBMS design, we should note that there are subtle problems **lurking** within the rollback process. The rolling back of one transaction might affect database entries that have been used by other transactions. For example, the transaction being rolled back might have updated an account balance, and another transaction might have already based its activities on this updated value. This might mean that these additional transactions must also be rolled back, which might

¹ *At this point the DBMS becomes committed to the transaction in the sense that it accepts the responsibility of guaranteeing that the transaction's activities will be reflected in the database.*: 在这个点上, 数据库管理系统在以下意义上向该事务作出承诺: 它接受保证该事务的活动将在数据库中得到反映的责任。在本句中, 第一个that引导的从句系同位语从句, 其本位语是sense; 第二个that引导的从句属于宾语从句, 作guaranteeing的宾语。

² *If problems should arise before a transaction has reached its commit point*: 在一个事务达到其提交点之前万一出现问题。本句中的should表示语气较强的假设。

³ *to roll back (undo) the activities actually performed by the transaction*: 回滚(撤销)该事务实际上已经进行的活动。

cascading

/ kæ'skeidiŋ /

a. 级联的; 层叠的

cascading rollback

级联回滚

flux / flʌks /

n. (不断的) 变动;

波动

update / 'ʌpdeɪt /

n. 更新; 修改

entirety / in'taiəti /

n. 全部; 整体

interweave

/ ,ɪntə'wi:v /

v. (使) 交织

shared lock

共享锁

adversely affect still other transactions. The result is the problem known as **cascading rollback**.

III. Locking

We now consider the problem of a transaction being executed while the database is in a state of **flux** from another transaction,¹ a situation that can lead to inadvertent interaction between the transactions and produce erroneous results. For instance, the problem known as the incorrect summary problem can arise if one transaction is in the middle of transferring funds from one account to another when another transaction tries to compute the total deposits in the bank. This could result in a total that is either too large or too small depending on the order in which the transfer steps are performed. Another possibility is known as the lost **update** problem, which is exemplified by two transactions, each of which makes a deduction from the same account. If one transaction reads the account's current balance at the point when the other has just read the balance but has not yet calculated the new balance, then both transactions will base their deductions on the same initial balance. In turn, the effect of one of the deductions will not be reflected in the database.

To solve such problems, a DBMS could force transactions to execute in their **entirety** on a one-at-a-time basis² by holding each new transaction in a queue until those preceding it have completed. But a transaction often spends a lot of time waiting for mass storage operations to be performed. By **interweaving** the execution of transactions, the time during which one transaction is waiting can be used by another transaction to process data it has already retrieved. Most large database management systems therefore contain a scheduler to coordinate time-sharing among transactions in much the same way that a time-sharing operating system coordinates interweaving of processes.

To guard against such anomalies as the incorrect summary problem and the lost update problem, these schedulers incorporate a locking protocol in which the items within a database that are currently being used by some transaction are marked as such.³ These marks are called locks; marked items are said to be locked. Two types of locks are common—**shared locks**

¹ *the database is in a state of flux from another transaction*: 数据库因另一事物而处于变迁状态。

² *a DBMS could force transactions to execute in their entirety on a one-at-a-time basis*: 数据库管理系统可以强制事务一次完整地执行一个。

³ *these schedulers incorporate a locking protocol in which the items within a database that are currently being used by some transaction are marked as such*: 这些调度程序都包含了一个锁定协议。在该协议中, 数据库中当前正在被某个事务使用的项目都要加以说明这种状态的标记。

exclusive lock排它锁,
互斥(型)锁
data item
数据项

and **exclusive locks**. They correspond to the two types of access to data that a transaction might require—shared access and exclusive access.¹ If a transaction is not going to alter a **data item**, then it requires shared access, meaning that other transactions are also allowed to view the data. However, if the transaction is going to alter the item, it must have exclusive access, meaning that it must be the only transaction with access to that data.

In a locking protocol, each time a transaction requests access to a data item, it must also tell the DBMS the type of access it requires. If a transaction requests shared access to an item that is either unlocked or locked with a shared lock, that access is granted and the item is marked with a shared lock. If, however, the requested item is already marked with an exclusive lock, the additional access is denied. If a transaction requests exclusive access to an item, that request is granted only if the item has no lock associated with it. In this manner, a transaction that is going to alter data protects that data from other transactions by obtaining exclusive access, whereas several transactions can share access to an item if none of them are going to change it. Of course, once a transaction is finished with an item, it notifies the DBMS, and the associated lock is removed.

Various algorithms are used to handle the case in which a transaction's access request is rejected. One algorithm is that the transaction is merely forced to wait until the requested item becomes available. This approach, however, can lead to deadlock, since two transactions that require exclusive access to the same two data items could block each other's progress if each obtains exclusive access to one of the items and then insists on waiting for the other. To avoid such deadlocks, some database management systems give priority to older transactions. That is, if an older transaction requires access to an item that is locked by a younger transaction, the younger transaction is forced to release all of its data items, and its activities are rolled back (based on the log). Then, the older transaction is given access to the item it required, and the younger transaction is forced to start again. If a younger transaction is repeatedly preempted, it will grow older in the process and ultimately become one of the older transactions with high priority.² This protocol, known as the wound-wait protocol (old transactions

¹ *They correspond to the two types of access to data that a transaction might require—shared access and exclusive access.*: 它们对应于事务可能需要的两种数据访问类型——共享访问和互斥访问。

² *If a younger transaction is repeatedly preempted, it will grow older in the process and ultimately become one of the older transactions with high priority.*: 如果一个较年轻的事务一再被抢先, 它将在这一过程中变得较老, 并最终成为一个具有高优先级的较老事务。

would young transactions, young transactions wait for old ones), ensures that every transaction will ultimately be allowed to complete its task.

Exercises

I. Fill in the blanks with the information given in the text:

1. The _____ point is the point at which all the steps in a transaction have been recorded in the log of the DBMS.
2. A large DBMS often contain a _____ to coordinate time-sharing among transactions in much the same way that a time-sharing operating system coordinates interweaving of processes.
3. The items within a database that are currently being used by some transaction are marked with locks. There are two common types of locks: _____ locks and _____ locks.
4. With the _____ protocol, if an older transaction requires access to an item that is locked by a younger transaction, the younger transaction is forced to release all of its data items, and its activities are rolled back.

II. Translate the following terms or phrases from English into Chinese and vice versa:

- | | |
|-------------------------------|------------|
| 1. nonvolatile storage system | 6. 共享锁 |
| 2. equipment malfunction | 7. 数据库实现 |
| 3. wound-wait protocol | 8. 级联回滚 |
| 4. exclusive lock | 9. 数据项 |
| 5. database integrity | 10. 分时操作系统 |

Section C

Data Mining

I. Introduction

data mining

数据挖掘

terabyte / 'terəbait /

n. 太字节, 万亿字节

petabyte / 'petəbait /

n. 拍字节, 千万亿字节

gigantic

/ dʒaɪ'ɡæntɪk /

a. 巨大的, 庞大的

data set

数据集

remote sensing

遥感

surveillance

/ sə:'veɪləns, sə- /

n. 监视

backbone

/'bækbəʊn /

n. 骨干(网),

基于(网)

search engine

搜索引擎

blog / blɒg /

n. 博客, 网志, 网络日志 (weblog 的缩略)

Web community

Web 社区,

网络社区

social network

社交网络

"We are living in the information age" is a popular saying; however, *we are actually living in the data age*. **Terabytes** or **petabytes** of data pour into our computer networks, the World Wide Web, and various data storage devices every day from business, society, science and engineering, medicine, and almost every other aspect of daily life. This explosive growth of available data volume is a result of the computerization of our society and the fast development of powerful data collection and storage tools. Businesses worldwide generate **gigantic data sets**, including sales transactions, stock trading records, product descriptions, sales promotions, company profiles and performance, and customer feedback. Scientific and engineering practices generate high orders of petabytes of data in a continuous manner,¹ from **remote sensing**, process measuring, scientific experiments, system performance, engineering observations, and environment **surveillance**.

Global **backbone** telecommunication networks carry tens of petabytes of data traffic every day. The medical and health industry generates tremendous amounts of data from medical records, patient monitoring, and medical imaging. Billions of Web searches supported by **search engines** process tens of petabytes of data daily. Communities and social media have become increasingly important data sources, producing digital pictures and videos, **blogs**, **Web communities**, and various kinds of **social networks**. The list of sources that generate huge amounts of data is endless.

This explosively growing, widely available, and gigantic body of data makes our time truly the data age. Powerful and versatile tools are badly needed to automatically uncover valuable information from the tremendous amounts of data and to transform such data into organized knowledge. This necessity has led to the birth of data mining. Data mining has and will continue to make great strides in our journey from the data age toward the

¹ *Scientific and engineering practices generate high orders of petabytes of data in a continuous manner*: 科学与工程实践以持续不断的方式产生高阶拍字节数据。

analogously

/ə'neɪləgəsli/

ad. 相似地, 类似地

nugget /'nʌɡɪt/n. 小块; 天然金块;
有价值的东西**misnomer**

/ˌmɪs'neɪmə/

n. 错误(或使用不当)的名字(或名称)

extraction

/ɪk'strækʃən/

n. 提取, 抽取,
析取**archaeology**

/ˌɑːki'ɒlədʒi/

n. 考古学

dredge /dredʒ/v. 疏浚; 挖掘;
采捞**synonym** /'sɪnənim/

n. 同义词

iterative

/ɪ'terətɪv, -reɪt-/

a. 重复的; 迭代的

coming information age.

II. Definition of Data Mining

To refer to the mining of gold from rocks or sand, we say *gold mining* instead of rock or sand mining. **Analogously**, data mining should have been more appropriately named “knowledge mining from data,” which is unfortunately somewhat long. However, the shorter term, *knowledge mining* may not reflect the emphasis on mining from large amounts of data. Nevertheless, mining is a vivid term characterizing the process that finds a small set of precious **nuggets** from a great deal of raw material. Thus, such a **misnomer** carrying both “data” and “mining” became a popular choice. In addition, many other terms have a similar meaning to data mining—for example, knowledge mining from data, knowledge extraction, data/pattern analysis, data **archaeology**, and data **dredging**.

Many people treat data mining as a **synonym** for another popularly used term, **knowledge discovery from data**, or **KDD**, while others view data mining as merely an essential step in the process of knowledge discovery. The knowledge discovery process is shown in Figure 6C-1 as an **iterative** sequence of the following steps:

1. **Data cleaning** (to remove noise¹ and inconsistent data)
2. **Data integration** (where multiple data sources may be combined)
3. **Data selection** (where data relevant to the analysis task are retrieved from the database)
4. **Data transformation** (where data are transformed and consolidated into forms appropriate for mining by performing summary or **aggregation** operations²)
5. **Data mining** (an essential process where intelligent methods are applied to **extract data patterns**)
6. **Pattern evaluation** (to identify the truly interesting patterns representing knowledge based on *interestingness measures*)
7. **Knowledge presentation** (where visualization and knowledge representation techniques are used to present mined knowledge to users)

aggregation

/ˌæɡrɪ'geɪʃən/

n. 聚集, 聚合,
集合**extract** /ɪk'strækt/v. 提取, 抽取,
析取**data pattern**

数据模式

¹ to remove noise: 消除数据噪声。

² data are transformed and consolidated into forms appropriate for mining by performing summary or aggregation operations: 通过进行汇总或聚合操作, 将数据变换与合并成适合挖掘的形式。

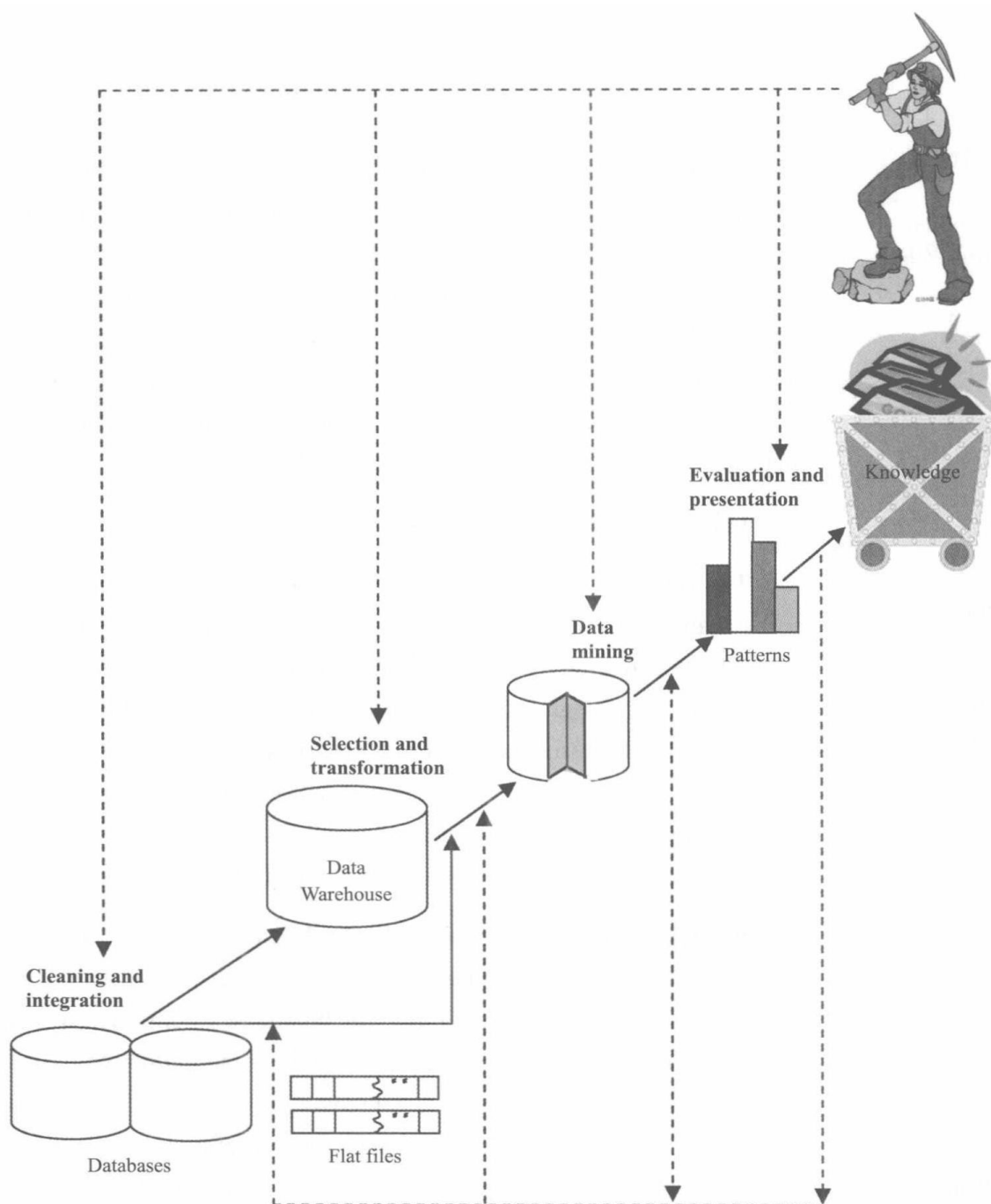


Figure 6C-1: Data Mining as a Step in the Process of Knowledge Discovery

warehouse

/ˈweəhaʊs/

n. 仓库, 货栈

data warehouse

数据仓库

warehousing

/ˈweəhaʊzɪŋ/

n. 存库, 入栈

knowledge base

知识库

A popular trend in the information industry is to perform data cleaning and data integration as a preprocessing step, where the resulting data are stored in a **data warehouse**. Sometimes data transformation and consolidation are performed before the data selection process, particularly in the case of data **warehousing**. Steps 1 through 4 are different forms of data preprocessing, where data are prepared for mining. The data mining step may interact with the user or a **knowledge base**. The interesting patterns are presented to the user and may be stored as new knowledge in

the knowledge base.

albeit /ɔ:l'bi:it/
conj. 尽管, 即使
milieu
/ 'mi:lje:, mi:'lje: /
n. (周围) 环境

The preceding view shows data mining as one step in the knowledge discovery process, **albeit** an essential one because it uncovers hidden patterns for evaluation. However, in industry, in media, and in the research **milieu**, the term *data mining* is often used to refer to the entire knowledge discovery process (perhaps because the term is shorter than *knowledge discovery from data*). Therefore, we adopt a broad view of data mining functionality: **Data mining** is the *process* of discovering interesting patterns and knowledge from *large* amounts of data. The data sources can include databases, data warehouses, the Web, other information repositories, or data that are streamed into the system dynamically¹.

III. Data Mining Trends

challenging
/ 'tʃælɪndʒɪŋ /
a. 挑战性的
interactive
/ .ɪntə'æktɪv /
a. 交互(式)的

The diversity of data, data mining tasks, and data mining approaches poses many **challenging** research issues in data mining. The development of efficient and effective data mining methods, systems and services, and **interactive** and integrated data mining environments is a key area of study. The use of data mining techniques to solve large or sophisticated application problems is an important task for data mining researchers and data mining system and application developers. This section describes some of the trends in data mining that reflect the **pursuit** of these challenges.

pursuit / pe'sju:t /
n. 追求; 追赶

- **Application exploration:** Early data mining applications put a lot of effort into helping businesses gain a competitive edge². The exploration of data mining for businesses continues to expand as e-commerce and e-marketing have become mainstream in the retail industry. Data mining is increasingly used for the exploration of applications in other areas such as web and text analysis, financial analysis, industry, government, **biomedicine**, and science. Emerging application areas include data mining for **counterterrorism** and mobile (wireless) data mining. Because generic data mining systems may have limitations in dealing with application-specific problems, we may see a trend toward the development of more application-specific data mining systems and tools, as well as invisible data mining functions embedded in various kinds of services.

biomedicine
/ .baɪəu'medɪsɪn /
n. 生物医学
counterterrorism
/ 'kauntə,terərɪzəm /
n. 反恐怖主义,
反恐

¹ *data that are streamed into the system dynamically*: 动态地流式输入系统的数据。

² *gain a competitive edge*: 取得竞争优势。

scalable /'skeiləbl /

a. 可缩放的, 可伸缩的

cloud computing

云计算

seamless /'si:mlis /

a. 无缝的

unified /'ju:nifaɪd /

a. 统一的

scalability

/,skeilə'biliti /

n. 可缩放性,

可伸缩性

multidimensional

/,mʌltidi'menʃ ənəl,

-daɪ'm- /

a. 多维的

ubiquitous

/ju:'bikwɪtəs /

a. 普遍存在的, 无所不在的

spatiotemporal

/,speɪʃiəʊ'tempərəl /

a. 时空的

cyber-physical

/,saɪbə'fɪzɪkəl /

a. 信息物理的

cellular /'seljʊlə /

a. 蜂窝状的, 多孔的

cellular phone

蜂窝电话, 移动电话, 手机

real time 实时

- **Scalable and interactive data mining methods:** In contrast with traditional data analysis methods, data mining must be able to handle huge amounts of data efficiently and, if possible, interactively. Because the amount of data being collected continues to increase rapidly, scalable algorithms for individual and integrated data mining functions become essential. One important direction toward improving the overall efficiency of the mining process while increasing user interaction is **constraint-based mining**. This provides users with added control by allowing the specification and use of constraints to guide data mining systems in their search for interesting patterns and knowledge.
- **Integration of data mining with search engines, database systems, data warehouse systems, and cloud computing systems:** Search engines, database systems, data warehouse systems, and cloud computing systems are mainstream information processing and computing systems. It is important to ensure that data mining serves as an essential data analysis component that can be smoothly integrated into such an information processing environment. A data mining subsystem/service should be tightly coupled with such systems as a **seamless, unified** framework or as an invisible function. This will ensure data availability, data mining portability, **scalability**, high performance, and an integrated information processing environment for **multidimensional** data analysis and exploration.
- **Mining social and information networks:** Mining social and information networks and link analysis are critical tasks because such networks are **ubiquitous** and complex. The development of scalable and effective knowledge discovery methods and applications for large numbers of network data is essential.
- **Mining spatiotemporal, moving-objects, and cyber-physical systems:** Cyber-physical systems as well as spatiotemporal data are mounting rapidly due to the popular use of **cellular phones**, GPS, sensors, and other wireless equipment. There are many challenging research issues realizing **real-time** and effective knowledge discovery with such data.¹

¹ *There are many challenging research issues realizing real-time and effective knowledge discovery with such data.*: 在实现这种数据的实时有效知识发现方面存在许多具有挑战性的研究问题。

biomedical

/ˌbaɪəʊ'medɪkəl/

a. 生物医学的

warrant

/'wɒrənt; 'wɔː-/

v. 证明...是正当
(或有理); 应受**high-dimensional****microarray**

高维微数组

biological pathway

生物学通路

bulky /'bʌlki/a. 体积大的, 庞大的;
又大又笨的**robustness**

/rəʊ'bʌstnis, 'rəʊbʌst-/

n. 健壮性, 坚固性

buggy /'bʌgi/a. 有(程序)错误的,
有故障的**disclose** /dis'kləʊz/v. 使显露; 泄露,
透露**outlier** /'aʊtlaɪə/n. 处于外面的东西;
外面部分**eventual**

/i'ventʃʊəl/

a. 最终发生的, 最后的

vigor /'vɪɡə/

n. 体力; 精力; 活力

data stream

数据流

- **Mining multimedia, text, and web data:** Mining such kinds of data is a recent focus in data mining research. Great progress has been made, yet there are still many open issues to be solved.
- **Mining biological and biomedical data:** The unique combination of complexity, richness, size, and importance of biological and biomedical data warrants special attention in data mining. Mining DNA¹ and protein sequences, mining **high-dimensional microarray** data, and **biological pathway** and network analysis are just a few topics in this field. Other areas of biological data mining research include mining biomedical literature, link analysis across heterogeneous biological data, and information integration of biological data by data mining.
- **Data mining with software engineering and system engineering:** Software programs and large computer systems have become increasingly **bulky** in size and sophisticated in complexity, and tend to originate from the integration of multiple components developed by different implementation teams. This trend has made it an increasingly challenging task to ensure software **robustness** and reliability. The analysis of the executions of a **buggy** software program is essentially a data mining process—tracing the data generated during program executions may **disclose** important patterns and **outliers** that could lead to the **eventual** automated discovery of software bugs. We expect that the further development of data mining methodologies for software/system debugging will enhance software robustness and bring new **vigor** to software/system engineering.
- **Visual and audio data mining:** Visual and audio data mining is an effective way to integrate with humans' visual and audio systems and discover knowledge from huge amounts of data. A systematic development of such techniques will facilitate the promotion of human participation for effective and efficient data analysis.
- **Distributed data mining and real-time data stream mining:** Traditional data mining methods, designed to work at a centralized location, do not work well in many of the distributed computing

¹ DNA: 脱氧核糖核酸 (deoxyribonucleic acid 的缩略)。

local area network

局域网

sensor network

传感器网络, 传感网, 感知网

stream data

流数据

intrusion

/ in'tru:ʒən /

n. 侵入; 打扰

abundance

/ ə'bʌndəns /

n. 大量, 丰富, 充足

confidential

/ .kənfi'denʃ əl /

a. 秘密的, 机密的

technologist

/ tek'nɒlədʒist /

n. 技术专家; 〈美〉技术员, 技师

environments present today (e.g., the Internet, intranets, **local area networks**, high-speed wireless networks, **sensor networks**, and cloud computing). Advances in distributed data mining methods are expected. Moreover, many applications involving **stream data** (e.g., e-commerce, Web mining, stock analysis, **intrusion** detection, mobile data mining, and data mining for counterterrorism) require dynamic data mining models to be built in real time. Additional research is needed in this direction.

- **Privacy protection and information security in data mining:** An **abundance** of personal or **confidential** information available in electronic forms, coupled with increasingly powerful data mining tools, poses a threat to data privacy and security. Growing interest in data mining for counterterrorism also adds to the concern. Further development of privacy-preserving data mining methods is foreseen. The collaboration of **technologists**, social scientists, law experts, governments, and companies is needed to produce a rigorous privacy and security protection mechanism for data publishing and data mining.

Exercises

I. Fill in the blanks with the information given in the text:

1. Data mining is the process of discovering interesting _____ and knowledge from large amounts of _____.
2. Many people treat data mining as a synonym for _____ discovery from data.
3. Traditional data mining methods, designed to work at a _____ location, do not work well in many of the _____ computing environments present today.
4. Cyber-physical systems as well as spatiotemporal data are mounting rapidly due to the popular use of cellular phones, GPS, sensors, and other _____ equipment.

II. Translate the following terms or phrases from English into Chinese and vice versa:

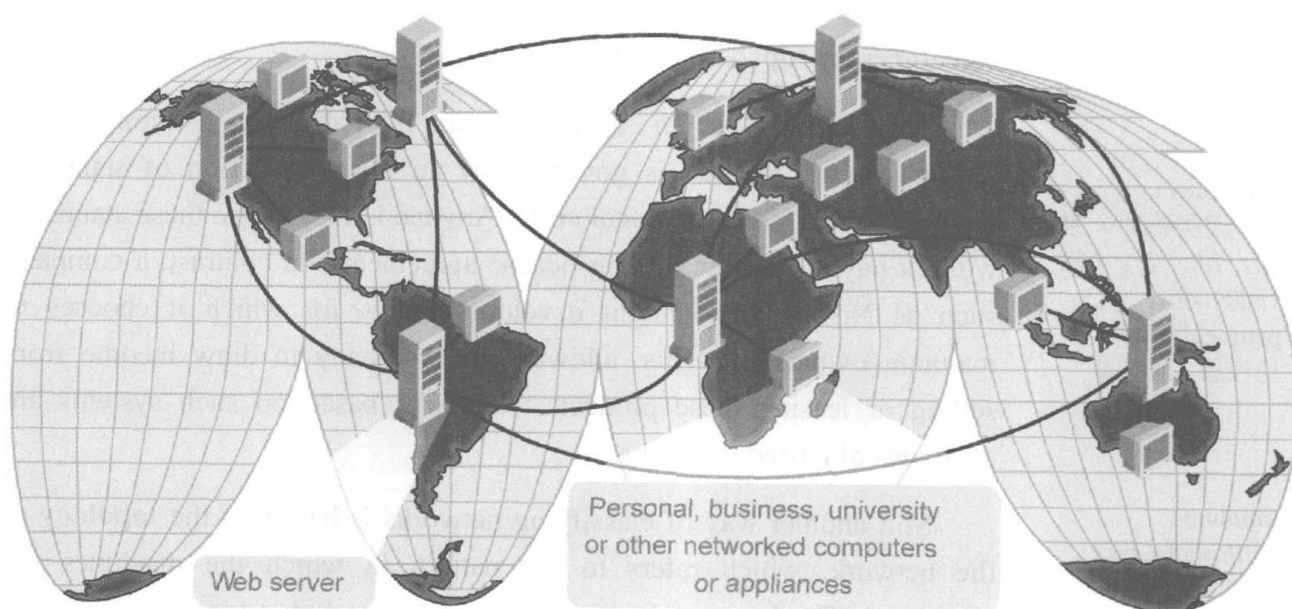
- | | |
|-------------------|---------|
| 1. data set | 5. 数据挖掘 |
| 2. data warehouse | 6. 社交网络 |
| 3. Web community | 7. 软件错误 |
| 4. sensor network | 8. 实时 |

Unit 7 Computer Network

(计算机网络)

Section A

Network Fundamentals



The need to share information and resources among different computers has led to linked computer systems, called networks, in which computers are connected so that data can be transferred from machine to machine. In these networks, computer users can exchange messages and share resources—such as printing capabilities, software packages, and data storage facilities—that are scattered throughout the system. The underlying software required to support such applications has grown from simple utility packages into an expanding system of network software that provides a sophisticated network wide infrastructure. In a sense, network software is evolving into a network wide operating system.

utility package

实用软件包,
公用程序包

metropolitan

/ˌmetrəˈpɒlɪtən/

a. 大城市的,
大都会的

**metropolitan area
network**

城域网

I. Network Classifications

A computer network is often classified as being either a local area network (LAN), a metropolitan area network (MAN), or a wide area network (WAN). A LAN normally consists of a collection of computers in a

single building or building complex. For example, the computers on a university campus or those in a manufacturing plant might be connected by a LAN. A MAN is a network of intermediate size, such as one spanning a local community. A WAN links machines over a greater distance—perhaps in neighboring cities or on opposite sides of the world.

Another means of classifying networks is based on whether the network's internal operation is based on designs that are in the public domain or on **innovations** owned and controlled by a particular entity¹ such as an individual or a corporation. A network of the former type is called an open network; a network of the latter type is called a closed, or sometimes a **proprietary**, network.

The Internet is an open system. In particular, communication throughout the Internet is governed by an open collection of standards known as the TCP/IP **protocol suite**. Anyone is free to use these standards without paying fees or signing license agreements. In contrast, a company such as Novell, Inc.² might develop systems for which it chooses to maintain ownership rights, allowing the company to draw income from selling or leasing these products. Networks based on such systems are examples of closed networks.

Still another way of classifying networks is based on the **topology** of the network, which refers to the pattern in which the machines are connected. The **bus topology**, **ring topology**, and **star topology** are three popular topologies. Of these, the star network is perhaps the oldest, having evolved from the paradigm of a large central computer serving many users. As the simple terminals employed by these users grew into small computers themselves, a star network emerged. A point to emphasize is that the connections between machines in a network do not need to be physical. Wireless networks, using radio broadcast technology, are becoming quite common.

II. Protocols

For a network to function reliably, it is important to establish rules by

innovation

/ˌɪnəʊ'veɪʃən/

n. 革新, 创新

proprietary

/prəʊ'praɪətəri/

a. 专有的, 专用的;
专利的

suite /swi:t/

n. (同类物的) 系列, 组, 套; (软件的) 套件

protocol suite

协议组, 协议集

topology

/təʊ'pɒlədʒi, tɒ-/

n. 拓扑(结构),
布局

bus topology

总线拓扑结构

ring topology

环型拓扑结构

star topology

星型拓扑结构

¹ *whether the network's internal operation is based on designs that are in the public domain or on innovations owned and controlled by a particular entity*: 网络的内部运行是基于无专利权保护的设计还是基于特定实体所拥有和控制的革新。句中的 public domain 表示“(发明、商标等的)不受专利限制(或保护)”、“(书等的)不受版权限制(或保护)”。

² *Novell, Inc.*: Novell 股份有限公司, 网威公司, 一家专门从事网络操作系统、安全身份管理产品及应用整合与合作方案开发的美国高科技企业, 最初创立于 1979 年, 总部设在美国犹他州普罗沃。

vender /'vendə/
n. 卖主; 厂家,
厂商

which network activities are conducted. Such rules are called protocols. By developing and adopting protocol standards, **venders** are able to build products for network applications that are compatible with products from other **venders**. Thus, the development of protocol standards is an indispensable process in the development of networking technologies.

As an introduction to the protocol concept, let us consider the problem of coordinating the transmission of messages among computers in a network. Without rules governing this communication, all the computers might insist on transmitting messages at the same time or might fail to relay messages when that assistance is required.

token /'təukən/
n. 权标, 令牌
token ring
权标环, 令牌环

One approach to solving this problem is the **token ring** protocol, which was developed by IBM in the 1970s and continues to be a popular protocol in networks based on the ring topology. In this protocol, all the machines in the network transmit messages in only one common direction (Figure 7A-1), meaning that all messages sent over the network move around the ring in the same direction by being forwarded from computer to computer. When a message reaches its destination, the destination machine keeps a copy of it and forwards a copy on around the ring. When the forwarded copy reaches the originating computer, that machine knows that the message must have reached its destination and removes the message from the ring. Of course, this system depends on inter-machine cooperation. If a machine insists on constantly transmitting messages of its own rather than forwarding those of the other machines, nothing will be accomplished.

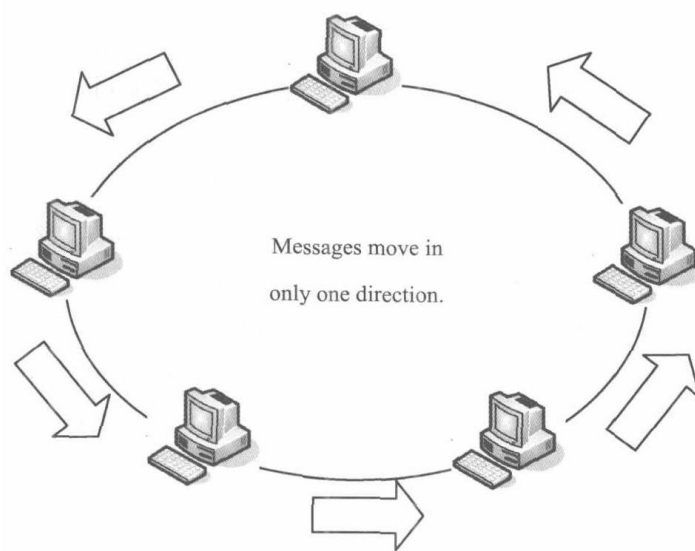


Figure 7A-1: Communication over a Ring Network

To solve this problem, a unique bit pattern, called a token, is passed around the ring. Possession of this token gives a machine the authority to transmit its own message; without the token, a machine is only allowed to forward messages. Normally, each machine merely relays the token in the same manner in which it relays messages. If, however, the machine receiving the token has messages of its own to introduce to the network, it transmits one message while holding the token. When this message has completed its cycle around the ring, the machine forwards the token to the next machine in the ring. Likewise, when the next machine receives the token, it can either forward the token immediately or transmit its own new message before sending the token on to the next machine. In this manner, each machine in the network has equal opportunity to introduce messages of its own as the token circles around the ring.

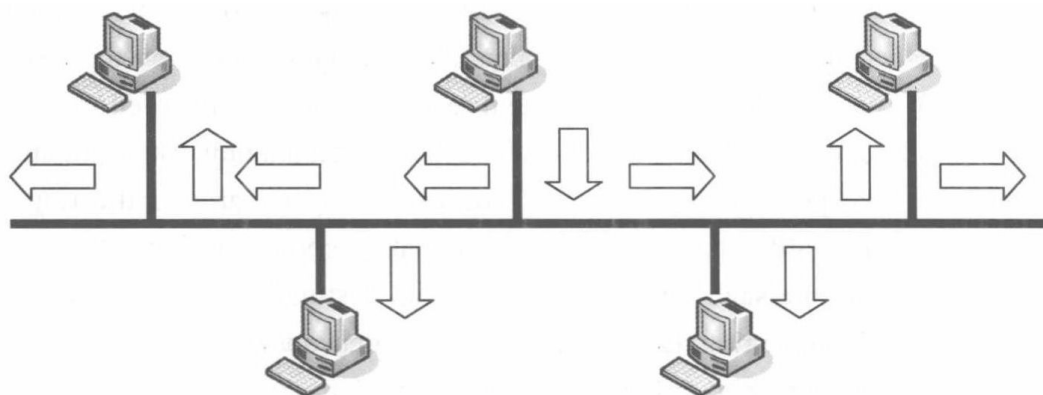


Figure 7A-2: Communication over a Bus Network

Another protocol for coordinating message transmission is used in bus topology networks that are based on the Ethernet protocol collection. In an Ethernet system, the right to transmit messages is controlled by the protocol known as **Carrier Sense Multiple Access with Collision Detection (CSMA/CD)**¹. This protocol dictates that each message be broadcast to all the machines on the bus (Figure 7A-2). Each machine monitors all the messages but keeps only those addressed to itself. To transmit a message, a machine waits until the bus is silent, and at this time it begins transmitting while continuing to monitor the bus. If another machine also begins transmitting, both machines detect the clash and pause for a brief random period of time before trying to transmit again. The result is a system similar

carrier /'kæriə/
n. 载波
carrier sense
载波检测,
载波监听

¹ CSMA/CD: 带有冲突检测的载波侦听多路访问(协议)(Carrier Sense Multiple Access with Collision Detection 的首字母缩略)。

to that used by a small group of people in a conversation. If two people start to talk at once, they both stop. The difference is that people might go through a series such as, "I'm sorry, what were you going to say?", "No, no. You go first," whereas under the CSMA/CD protocol each machine merely tries again.

III. Interprocess Communication

The various activities (or processes) executing on the different computers within a network (or even executing on the same machine via time sharing) must often communicate with each other to coordinate their actions and to perform their designated tasks. Such communication between processes is called **interprocess communication**.

**interprocess
communication**
进程间通信

A popular convention used for interprocess communication is the client/server model. This model defines the basic roles played by the processes as either a client, which makes requests of other processes, or a server, which satisfies the requests made by clients.

cluster /'klʌstə/
n. (人或物的) 群,
组; (果实、花等的)
串, 束, 簇
print server
打印服务器

An early application of the client/server model appeared in networks connecting all the computers in a **cluster** of offices. In this situation, a single, high-quality printer was attached to the network where it was available to all the machines in the network. In this case the printer played the role of a server (often called a **print server**), and the other machines were programmed to play the role of clients that sent print requests to the print server.

duplicate
/'dʒu:plikit/
a. 复制的; 副(本)
的; 重复的

Another early application of the client/server model was used to reduce the cost of disk storage while also removing the need for **duplicate** copies of records. Here one machine in a network was equipped with a high-capacity mass storage system (usually a magnetic disk) that contained all of an organization's records. Other machines on the network then requested access to the records as they needed them. Thus the machine that actually contained the records played the role of a server (called a **file server**), and the other machines played the role of clients that requested access to the files that were stored at the file server.

file server
文件服务器

peer-to-peer
/'pi:tə'piə/
a. 对等的

Today the client/server model is used extensively in network applications. However, the client/server model is not the only means of interprocess communication. Another model is the **peer-to-peer** (often abbreviated P2P) model, whose properties provide striking contrasts to the client/server model. Whereas the client/server model involves one process

instant messaging

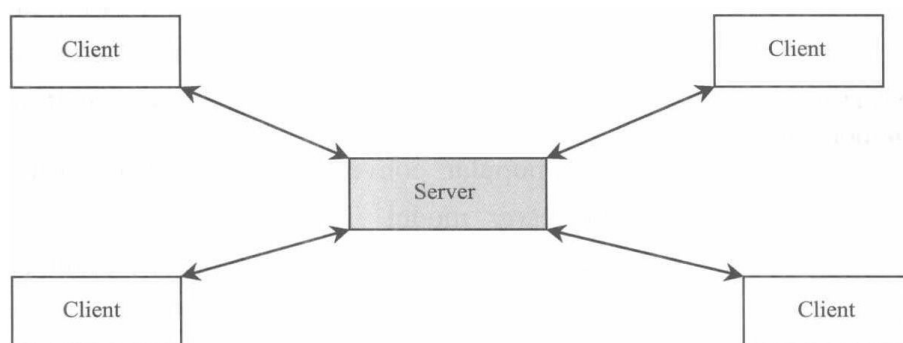
即时通信,

即时消息

checkers /'tʃekəz/

n. 〈美〉西洋跳棋

(the server) communicating with numerous others (clients), the peer-to-peer model involves two processes communicating as equals (Figure 7A-3). Moreover, whereas a server must execute continuously so that it is prepared to serve its clients at any time, the peer-to-peer model usually involves two processes that execute on a temporary basis. For example, applications of the peer-to-peer model include **instant messaging** in which two people carry on a written conversation over the Internet as well as situations in which people play games such as chess or **checkers**.



a) A server must be prepared to serve multiple clients at any time.



b) Peers communicate as equals on a one-to-one basis.

Figure 7A-3: The Client/Server Model Compared to the Peer-to-Peer Model

motion picture

〈美〉影片, 电影

legality /li'gæləiti/

n. 合法(性)

The peer-to-peer model is also a popular means of sharing files such as music recordings and **motion pictures** via the Internet (sometimes with questionable **legality**). In this case, individuals who are seeking particular items broadcast their desire over the Internet and are contacted by those who own the items of interest.¹ Then, the items are transferred between the two parties using the peer-to-peer model. This is in contrast to earlier approaches that applied the client/server model by establishing a central “distribution center” (the server) from which clients downloaded music recordings (or at least found sources for those items). The central server,

¹ In this case, individuals who are seeking particular items broadcast their desire over the Internet and are contacted by those who own the items of interest.: 在这种情况下, 寻求特定项目的个人在因特网上发布其需求, 而拥有他所感兴趣项目的人则会与他取得联系。

focal /'fəukəl/

a. 焦点的

focal point

焦点, 活动(或注意、兴趣等的)中心

dismantle

/dis'mæntl/

v. 解散; 拆开;

废除

terminology

/,tə:'mɪnələdʒi/

v. [总称] 术语

internet /'ɪntənət/

n. 互联网, 互连网

however, proved to be a **focal point** at which the music industry could enforce copyright laws, leading to the eventual **dismantling** of these music distribution centers. In contrast, the lack of a centralized base of operation obtained via the peer-to-peer model makes legal efforts to enforce copyright laws more difficult.

You might often read or hear the term *peer-to-peer network*, which is an example of how misuse of terminology can evolve when technical terms are adopted by the nontechnical community.¹ The term *peer-to-peer* refers to a system by which two processes communicate over a network (or **internet**²). It is not a property of the network (or internet). A process might use the peer-to-peer model to communicate with another process and later use the client/server model to communicate with another process over the same network. Thus, it would be more accurate to speak of communicating by means of the peer-to-peer model rather than communicating over a peer-to-peer network.

Exercises

I. Fill in the blanks with the information given in the text:

1. According to the geographical area covered, a computer network is often classified as either a LAN, a(n) _____, or a WAN.
2. A network can also be classified as either a(n) _____ network or a(n) _____ network according to whether its internal operation is based on designs in the public domain or on innovations owned and controlled by a particular entity.
3. A network can also be classified according to its topology. Among the most popular topologies are the _____ topology, ring topology, and star topology.
4. A popular protocol in networks based on the ring topology is the _____ ring protocol developed by IBM in the 1970s.
5. In a(n) _____ system, the right to transmit messages is

¹ an example of how misuse of terminology can evolve when technical terms are adopted by the nontechnical community: 当技术术语被非技术界采用时可能发生术语误用的一个例子。

² internet: 常译作“互联网(络)”,系 internetwork 的简写形式,指的是一组通过网关联结在一起的可能互不相同的计算机网络。注意 internet 与 Internet 的不同,后者常译作“因特网”、“国际互联网”。

controlled by the protocol known as CSMA/CD.

6. The _____ model for interprocess communication defines the basic roles played by the processes as either a client or a server.

7. The peer-to-peer model involves two processes communicating as _____ and usually executing on a(n) _____ basis.

8. Although it is often used, the term peer-to-peer _____ is not very appropriate.

II. Translate the following terms or phrases from English into Chinese and vice versa:

- | | |
|---------------------------------|-----------------|
| 1. file server | 11. 局域网 |
| 2. carrier sense | 12. 令牌环 |
| 3. protocol suite | 13. 无线网络 |
| 4. peer-to-peer model | 14. 封闭式网络 |
| 5. bus topology network | 15. 环型拓扑结构 |
| 6. inter-machine cooperation | 16. 客户机 / 服务器模型 |
| 7. Ethernet protocol collection | 17. 网络应用程序 |
| 8. proprietary network | 18. 进程间通信 |
| 9. utility package | 19. 打印服务器 |
| 10. star network | 20. 广域网 |

III. Fill in each of the blanks with one of the words given in the following list, making changes if necessary:

- | | | | |
|--------------------|------------------|--------------------|------------------|
| <i>space</i> | <i>memory</i> | <i>access</i> | <i>manage</i> |
| <i>maintenance</i> | <i>advantage</i> | <i>workstation</i> | <i>bulk</i> |
| <i>reduce</i> | <i>server</i> | <i>networked</i> | <i>expensive</i> |
| <i>remote</i> | <i>within</i> | <i>exchange</i> | <i>without</i> |

Computers can communicate with other computers through a series of connections and associated hardware called a network. The _____ of a network is that data can be _____ rapidly, and software and hardware resources, such as hard-disk _____ or printers, can be shared. Networks also allow _____ use of a computer by a user who cannot physically _____ the computer.

One type of network, a local area network (LAN), consists of several

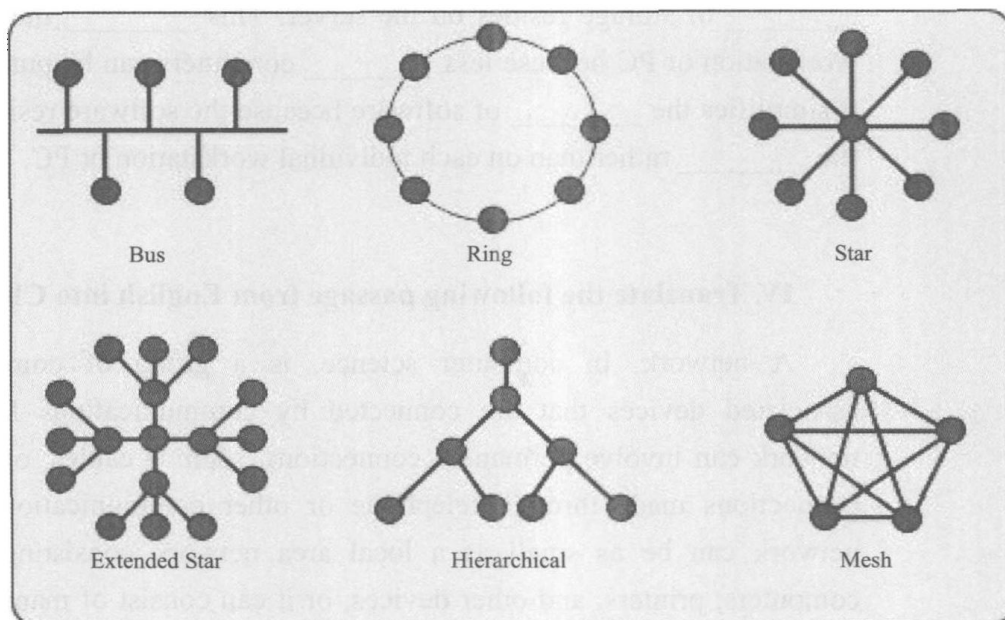
PCs or _____ connected to a special computer called a server, often _____ the same building or office complex. The server stores and _____ programs and data. A server often contains all of a _____ group's data and enables LAN workstations or PCs to be set up _____ large storage capabilities. In this *scenario* (方案), each PC may have "local" _____ (for example, a hard drive) specific to itself, but the _____ of storage resides on the server. This _____ the cost of the workstation or PC because less _____ computers can be purchased, and it simplifies the _____ of software because the software resides only on the _____ rather than on each individual workstation or PC.

IV. Translate the following passage from English into Chinese:

A network, in computer science, is a group of computers and associated devices that are connected by communications facilities. A network can involve permanent connections, such as cables, or temporary connections made through telephone or other communications links. A network can be as small as a local area network consisting of a few computers, printers, and other devices, or it can consist of many small and large computers distributed over a vast geographic area. Small or large, a computer network exists to provide computer users with the means of communicating and transferring information electronically. Some types of communication are simple user-to-user messages; others, of the type known as distributed processes, can involve several computers and the sharing of workloads or *cooperative* (合作的) efforts in performing a task.

Section B

A Guide to Network Topology



routing path

路由选择通路

familiarize

/fə'miliəraɪz/

v. 使熟悉; 使通晓

mesh /meʃ/

n. 网; 网格; 网状结构; 网状网络

A network topology is how computers, printers, and other devices are connected over a network. It describes the layout of wires, devices, and **routing paths**. Essentially there are six different common topologies you should **familiarize** yourself with: bus, ring, star, extended star, hierarchical, and **mesh**.

I. Bus Topology

The bus topology was fairly popular in the early years of networking. It's easy to set up—not to mention inexpensive. All devices on the bus topology are connected using a single cable. If you need help remembering how the bus topology operates, think of it as the route a bus takes throughout a city.

terminator

/ˈtɜːmɪneɪtə/

n. 终结器; 端子; 终结符

disrupt /dɪs'rʌpt/

v. 扰乱; 使中断

It is extremely important to note that both ends of the main cable need to be terminated. If there is no **terminator**, the signal will bounce back when it reaches the end. The result: a bunch of collisions and noise that will **disrupt** the entire network.

The bus topology is less common these days. In fact, this topology is

coaxial /ˌkəʊˈæksɪəl/

a. 同轴的

coaxial cable

同轴电缆

redundant

/rɪˈdʌndənt/

a. 冗余的

slack /slæk/

n. 松弛 (部分)

down /daʊn/

v. 击倒; 击落;

倒下

commonly used to network computers via **coaxial cable**—when's the last time you can say you've done that?

II. Ring Topology

The ring topology is a very interesting topology indeed. It is a lot more complex than it may seem—it looks like just a bunch of computers connected in a circle! But behind the scenes, the ring topology is providing a collision-free and **redundant** networking environment.¹

Note that since there is no end on a ring topology, no terminators are necessary. A frame travels along the circle, stopping at each node. If that node wants to transmit data, it adds destination address and data information to the frame. The frame then travels around the ring, searching for the destination node. When it's found, the data is taken out of the frame and the cycle continues.

But wait—it gets better! We have two types of ring topologies in networking: the one we just reviewed, and dual-ring topology. In a dual-ring topology, we use two rings instead of one. This creates a sense of redundancy so that if any point in the network fails, the second ring will (hopefully) be able to pick up the **slack**.² If both rings were to fail at separate locations, we can even use the opposite ring at each point to “patch” the **downed** node.³

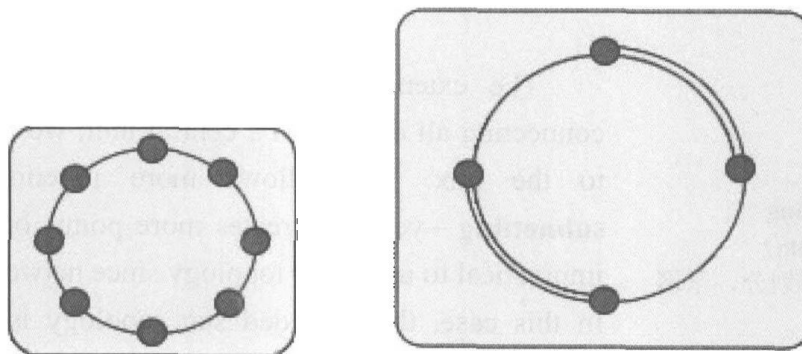


Figure 7B-1: The Ring and Dual-Ring Topology

¹ *But behind the scenes, the ring topology is providing a collision-free and redundant networking environment.*: 但是, 在幕后, 环型拓扑结构却提供着一个无冲突的冗余连网环境。

² *This creates a sense of redundancy so that if any point in the network fails, the second ring will (hopefully) be able to pick up the slack.*: 这在一定意义上产生了冗余, 因此, 如果网络中的任何点失灵, 第二个环将 (有望) 能够予以弥补。

³ *If both rings were to fail at separate locations, we can even use the opposite ring at each point to “patch” the downed node.*: 如果两个环分别在不同的点失灵, 我们甚至可以在每个点使用对面的环来“修补”发生故障的节点。本句前半部分是一个虚拟条件状语从句, 表示所述的情况发生的可能性不大。

In the diagram on the right in Figure 7B-1, you can see that although the outer ring and inner ring failed at separate parts of the network, thanks to redundancy, the network is still fully operational. This is generally more expensive to implement than other topologies—so it isn't as common as the star or extended star topology.

III. Star/Extended Star Topology

One of the most popular topologies for Ethernet LANs is the star and extended star topology. It is easy to set up, it's relatively cheap, and it creates more redundancy than the bus topology.

The star topology works by connecting each node to a central device. This central connection allows us to have a fully functioning network even when other devices fail. The only real threat to this topology is that if the central device goes down, so does the entire network.

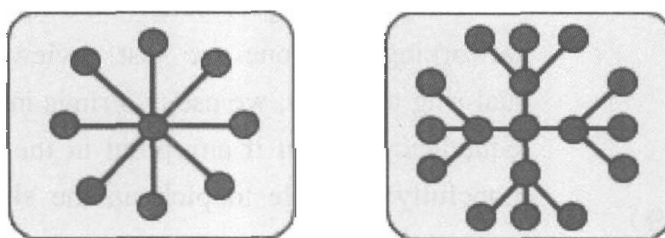


Figure 7B-2: The Star and Extended-Star Topology

The extended star topology is a bit more advanced. Instead of connecting all devices to a central unit, we have sub-central devices added to the mix. This allows more functionality for organization and **subnetting**—yet also creates more points of failure.¹ In many cases, it is impractical to use a star topology since networks can span an entire building. In this case, the extended star topology is all but necessary to prevent **degraded** signals.²

Whereas the star topology is better suited for small networks, the extended star topology is generally better for the larger ones.

subnetting

/ 'sʌb,netɪŋ /

n. 子网划分, 子网
组建

degraded

/ di'greɪdɪd /

a. 降级的, 退化的

¹ *This allows more functionality for organization and subnetting—yet also creates more points of failure.*: 这就在网络的组织与子网划分方面提供了更多功能——但也造成更多的故障点。

² *In this case, the extended star topology is all but necessary to prevent degraded signals.*: 在这种情况下, 为防止信号衰减, 扩展星型拓扑结构几乎是必需的。all but 意为“几乎”、“差不多”。

IV. Hierarchical Topology

The hierarchical topology is much like the star topology, except that it doesn't use a central node. Although Cisco¹ prefers to call this hierarchical, you may see it as instead referred to as the tree topology.

This type of topology suffers from the same centralization flaw as the star topology. If the device that is on top of the chain fails, consider the entire network down. Obviously this is impractical and not used a great deal in real applications.

V. Mesh Topology

If you haven't noticed, we've had a little problem with a fully redundant network. The dual-ring topology helped, but it wasn't perfect. If you are looking for a truly redundant network, look no further than the mesh topology. You will see two main types of mesh topology: full-mesh and partial-mesh.

The full-mesh topology connects every single node together. This will create the most redundant and reliable network around—especially for large networks. If any link fails, we (should) always have another link to send data through. So why don't we use it more often? Simple: how many wires would it take to link a computer to every device on a network of over 100 devices? Now multiply that for every device on the network—not a pleasant number is it? Obviously you should only use this in smaller networks. **Alternatively**, you could try a partial-mesh topology.

alternatively

/ɔ:l'te:nativli/

ad. 或者, 非此即彼

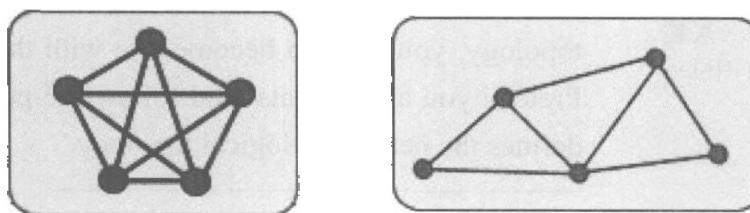


Figure 7B-3: The Full-Mesh and Partial-Mesh Topology

alternate

/ɔ:l'te:nit, 'ɔ:l'te-/

a. 供选择的, 供替换的; 备用的

The partial-mesh topology is much like the full-mesh, only we don't connect each device to every other device on the network. Instead, we only implement a few **alternate** routes. After all, what are the odds a network will fail in multiple times near the same device?²

¹ Cisco: 即 Cisco Systems, Inc., 思科系统有限公司, 1984 年成立于美国硅谷, 系全球网络和通信领域公认的领先厂商。思科 (Cisco) 的名字取自旧金山 (San Francisco)。

² After all, what are the odds a network will fail in multiple times near the same device?: 毕竟, 一个网络在同一个设备附近多次出故障的可能性有多大呢? 言外之意, 这种可能性不大。

Internet service provider
因特网服务提供商
router /'ru:tə/
n. 路由器

hybrid /'haibrid/
n. 混合物

endpoint
/'endpoint/
n. 端点

hub /hʌb/
n. (网络) 集线器;
(轮) 毂; (兴趣、
活动等的) 中心

You'll see the partial-mesh topology in backbone environments, since these are often vital networks that depend on redundancy to keep services running (such as an **Internet service provider**). The full-mesh topology is commonly seen in WANs between **routers**, yet also on smaller networks that depend on a redundant connection.

VI. Closing Comments

Keep in mind that network topology isn't limited to the above examples. There are **hybrids** and variations of the topologies mentioned above. Oddly enough, Cisco fails to categorize point-to-point topology in their course material—but don't worry, it's just a simple connection between two **endpoints**. Perhaps it was considered too simple to include in the course material—either way, make sure you commit the above topologies to memory.¹ You'll be expected to know them when exam day comes—not to mention it could save you from a disorganized mess of a network!

Another point to emphasize is that each network has a logical topology as well as a physical topology. The physical and logical topology of the network can be the same, but often they aren't. Think of the physical topology as the network layout you'd see if you looked at the physical network from outside. To understand the physical topology, you completely ignore how the data travels, and focus instead on the appearance of cables, **hubs**, and nodes that make up the network. The logical topology describes how data actually travels on the network. To understand the logical topology, you need to become one with the data traveling on the network. Pretend you are the data, and follow the path it takes. The path you follow defines the network's logical topology.

Exercises

I. Fill in the blanks with the information given in the text:

1. In the bus topology, a _____ should be installed at each end of the bus or main cable.
2. As compared with the star topology, the extended star topology adds

¹ *commit the above topologies to memory*: 记住上面的拓扑结构。commit sth to memory 表示“把…记住”、“把…背熟”。

_____ devices instead of connecting all devices to a central unit.

3. The hierarchical topology, also referred to as the _____ topology, is much like the star topology, except that it doesn't use a(n) _____ node.

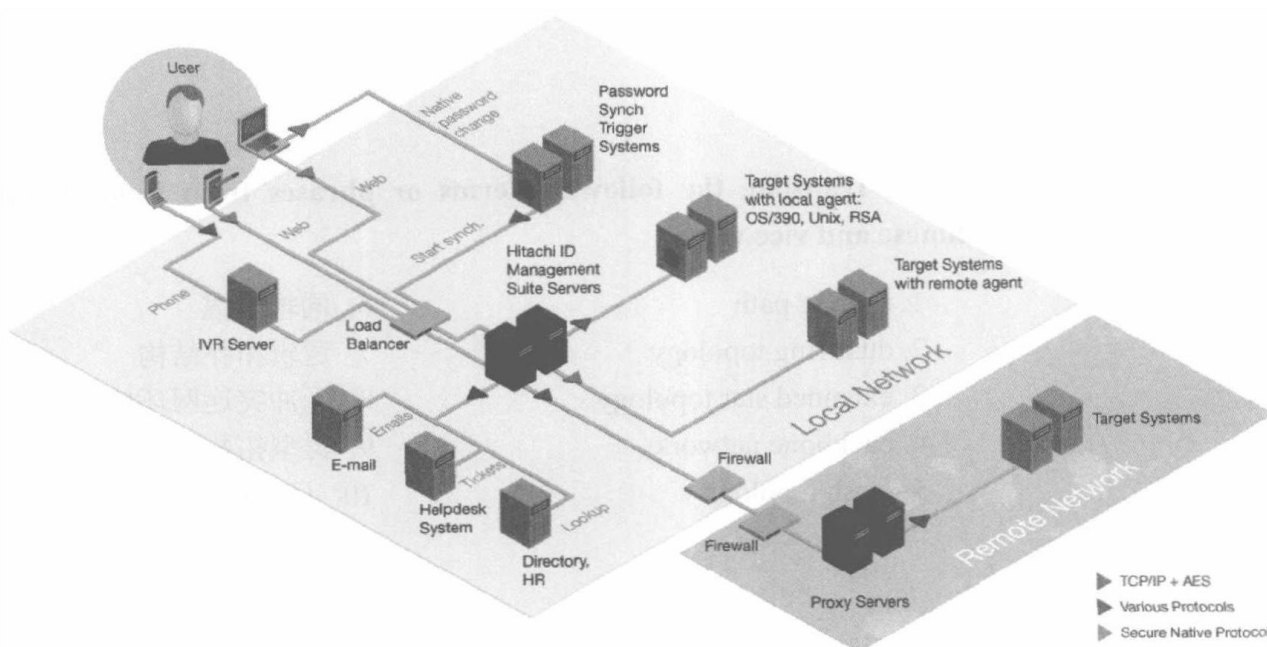
4. A truly redundant network can be created with the _____ topology.

II. Translate the following terms or phrases from English into Chinese and vice versa:

- | | |
|---------------------------|------------|
| 1. routing path | 6. 同轴电缆 |
| 2. dual-ring topology | 7. 逻辑拓扑结构 |
| 3. extended star topology | 8. 无冲突连网环境 |
| 4. backbone network | 9. 树型拓扑结构 |
| 5. mesh topology | 10. 目的地节点 |

Section C

Network Connecting Devices



We can divide computer networks into three broad categories: local area networks (LANs), metropolitan area networks (MANs), and wide area networks (WANs). The three network types can be connected using connecting devices. The interconnection of networks makes global communication from one side of the world to the other possible. Connecting devices can be divided into four types based on their functionality as related to the layers in the OSI model: **repeaters**, **bridges**, **routers**, and **gateways**. Repeaters and bridges typically connect devices in a network. Routers and gateways typically connect networks into **internetworks** (Figure 7C-1).

repeater /ri'pi:tə/
n. 中继器, 转发器
gateway /'geitwei/
n. 网关
internetwork
/,intə'netwə:k/
n. 互联网, 互连网

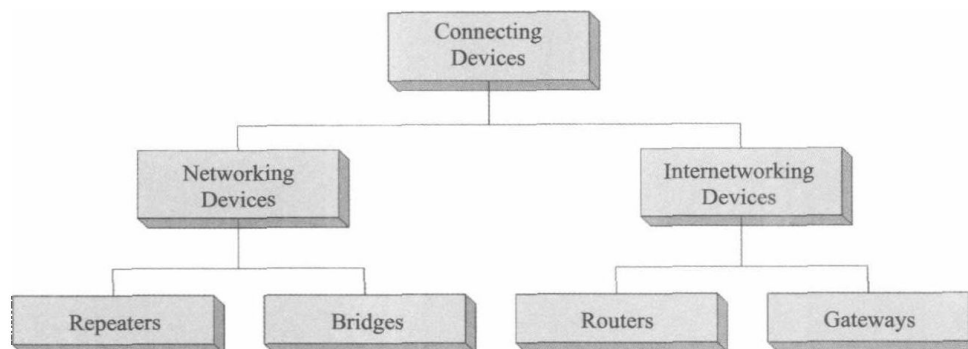


Figure 7C-1: Network Connecting Devices

A repeater is an electronic device that regenerates data. It extends the physical length of a network. As a signal is transmitted, it may lose strength, and a weak signal may be interpreted erroneously by a receiver. A repeater can regenerate the signal and send it to the rest of the network. Figure 7C-2 shows a network with and without a repeater.

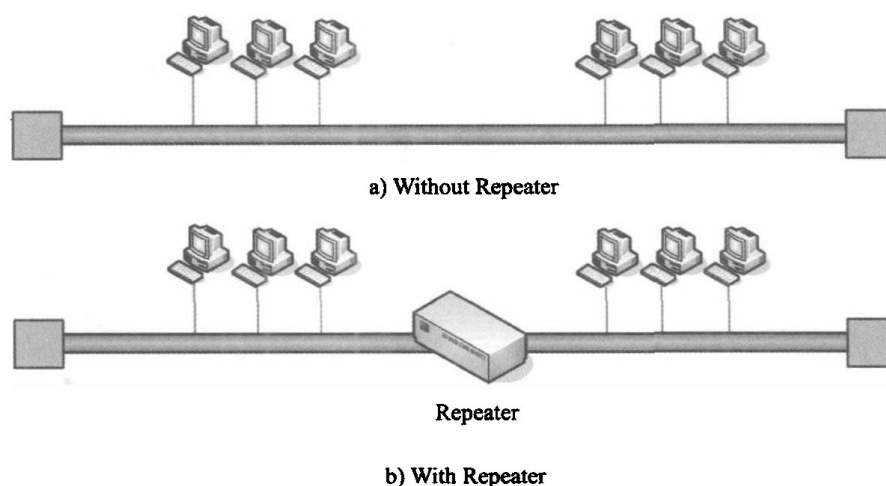


Figure 7C-2: Repeater

Repeaters operate only in the physical layer of the OSI model. They do not recognize physical or logical addresses. They simply regenerate every signal they receive. Repeaters, popular when the dominant topology was the bus topology, often connected two buses to increase the length of the network.

When a network uses a bus topology, the medium is shared between all stations. In other words, when a station sends a frame, the common bus is occupied by this one station, and no other station is allowed to send a frame (if it does, the two frames **collide**). This implies a **degradation** of performance. Stations need to wait until the bus is free. This is similar to an airport that has only one **runway**. When the runway is used by one aircraft, the other aircraft ready for takeoff must wait.

A bridge is a traffic controller. It can divide a long bus into smaller segments so that each segment is independent **trafficwise**.¹ A bridge installed between two segments can pass or block frames based on the destination address in the frame. If a frame originates in one segment and the destination is in the same segment, there is no reason for the frame to pass the bridge and go to the other segments. The bridge uses a table to

collide /kə'laɪd/
v. 冲突; 碰撞
degradation
/ˌdeɡreɪ'deɪʃən/
n. 降级, 退化
runway /'rʌnwei/
n. (机场的) 跑道

trafficwise
/'træfɪkwaɪz/
ad. 在交通方面

¹ A bridge is a traffic controller. It can divide a long bus into smaller segments so that each segment is independent trafficwise.: 网桥相当于交通管制员。它可以把一条长总线分成若干较小的段, 这样每一段就交通(这里指通信)而言都是独立的。

decide if the frame needs to be forwarded to another segment. With a bridge, two or more pairs of stations can communicate at the same time (Figure 7C-3).

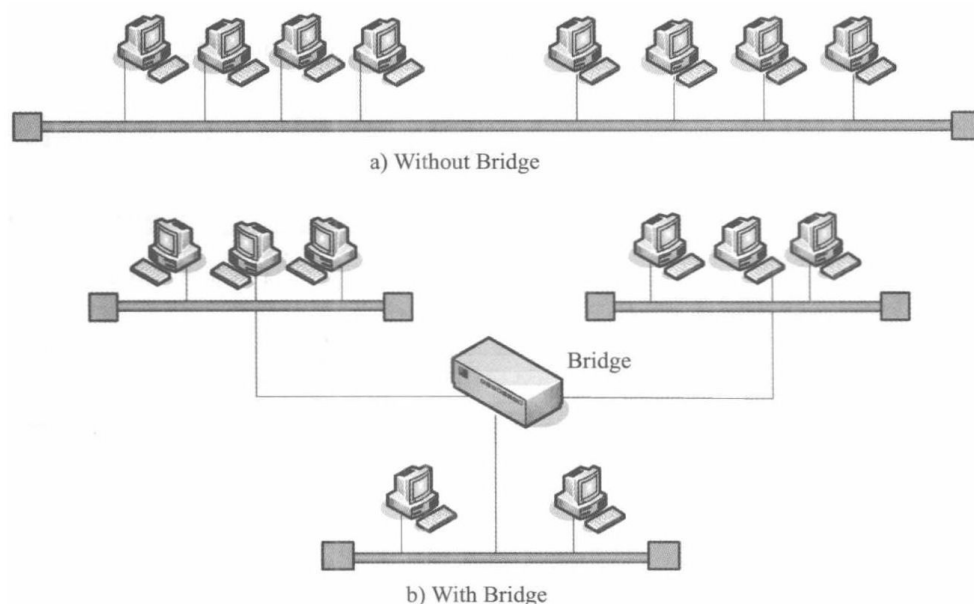


Figure 7C-3: Bridge

In addition to its traffic controlling duties, a bridge also functions as a repeater by regenerating the frame. As discussed previously, this means that a bridge operates at the physical layer. But because a bridge needs to interpret the address embedded in the frame to make filtering decisions, it also operates at the **data-link** layer of the OSI model.

data link
数据链路

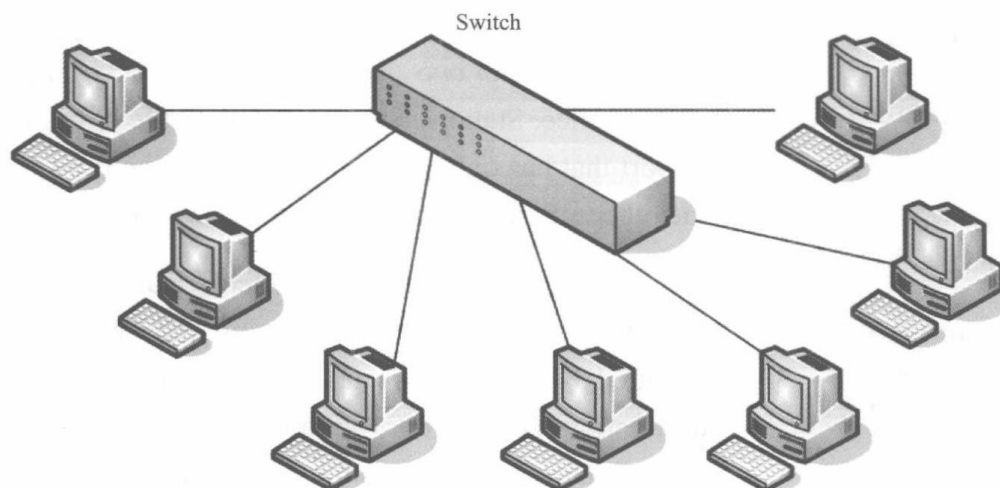


Figure 7C-4: Switch

In recent years, the need for better performance has led to the design of a new device referred to as a second-layer switch, which is simply a

sophisticated bridge with multiple interfaces. For example, a network with 20 stations can be divided into four segments using a four-interface bridge. Or the same network can be divided into 20 segments (with one station per segment) using a 20-interface switch. A switch in this case increases performance; a station that needs to send a frame sends it directly to the switch. The media are not shared; each station is directly connected to the switch (Figure 7C-4).

Routers are devices that connect LANs, MANs, and WANs. A router operates at the third layer of the OSI model. Whereas a bridge filters a frame based on the physical (data-link layer) address of the frame, a router **routes** a packet based on the logical (network layer) address of the packet.

Whereas a bridge may connect two segments of a LAN or two LANs belonging to the same organization, a router can connect two independent networks: a LAN to a WAN, a LAN to a MAN, a WAN to another WAN, and so on. The result is an internetwork (or an internet). The Internet (the unique global internet) that connects the whole world together is an example of an internetwork where many networks are connected together through routers. Figure 7C-5 shows an example of an internetwork.

route /ru:t/
v. (按特定路径)
发送, 传递

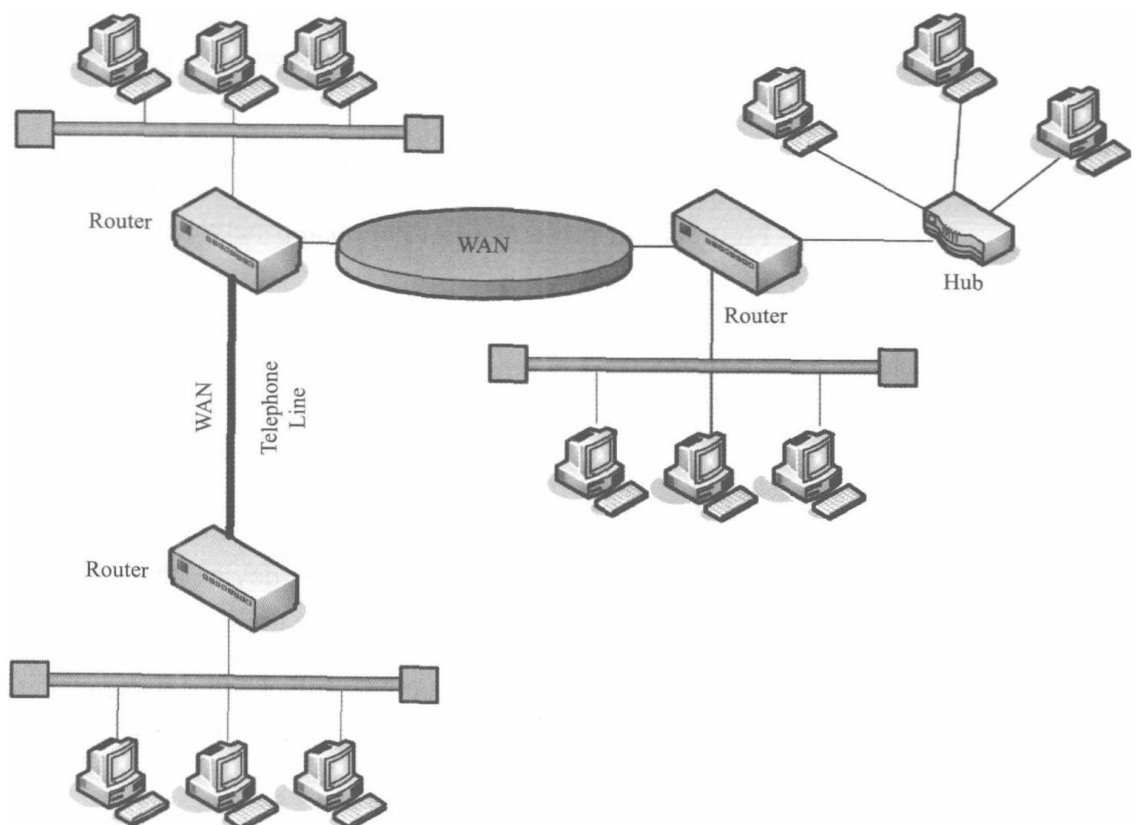


Figure 7C-5: Routers in an Internet

converter
/kən'veɪtə/
n. 转换器,
转换程序

interchangeable
/,ɪntə'tʃeɪndʒəbəl/
a. 可交换的, 可互
换的

Traditionally, a gateway is a connecting device that acts as a protocol **converter**. It allows two networks, each with a different set of protocols for all seven OSI layers, to be connected to each other and communicate. A gateway is usually a computer installed with the necessary software. The gateway understands the protocols used by each connected network and is therefore able to translate from one to another. For example, a gateway can connect a network using the AppleTalk¹ protocol to a network using the Novell Netware² protocol.

Today, however, the term gateway is used **interchangeably** with the term router. Some people refer to a gateway as a router, and others refer to a router as a gateway. The distinction between the two terms is disappearing.

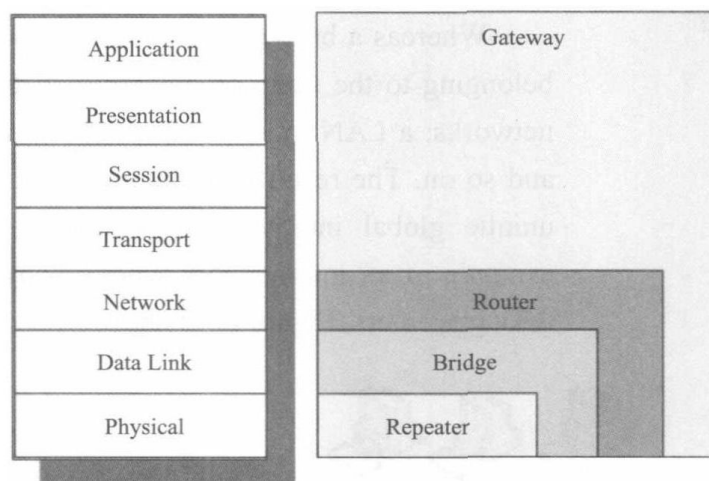


Figure 7C-6: Connecting Devices and the OSI Model

Figure 7C-6 presents the relationship between the connecting devices and the OSI model.

Exercises

I. Fill in the blanks with the information given in the text:

1. A(n) _____ is a connecting device that only regenerates the signal and does not recognize physical or logical addresses.
2. Whereas a repeater operates only in the physical layer of the OSI model, a bridge operates in both the physical layer and the _____ layer of the OSI model.

¹ AppleTalk: 美国苹果公司 Mac 机所用的网络协议之一。

² Novell Netware: 美国网威 (Novell) 公司网络操作系统。

3. A(n) _____ is usually a computer installed with the necessary software and allows two networks, each with a completely different set of protocols, to communicate.

4. The distinction between the two terms gateway and _____ is disappearing. They are now used interchangeably.

II. Translate the following terms or phrases from English into Chinese and vice versa:

- | | |
|----------------------------|------------|
| 1. destination address | 5. 数据链路层 |
| 2. performance degradation | 6. 协议转换器 |
| 3. four-interface bridge | 7. 开放式系统互连 |
| 4. common bus | 8. 物理地址 |

Unit 8 The Internet

(因特网)

Section A

The Internet



uppercase

/ˈʌpəˈkeɪs/

a. (字母) 大写的

The most notable example of an internet is the Internet (note the **uppercase I**), which originated from research projects going back to the early 1960s. The goal was to develop the ability to link a variety of computer networks so that they could function as a connected system that would not be disrupted by local disasters. Most of this original work was sponsored by the U.S. government through the Defense Advanced Research

undertaking

/ˌʌndə'teɪkɪŋ, 'ʌndət-/

n. 任务; 事业

Projects Agency (DARPA¹—pronounced “DAR-pa”). Over the years, the development of the Internet shifted from a defense project to an academic research project, and today it is largely a commercial **undertaking** that links a worldwide combination of WANs, MANs, and LANs involving millions of computers.

I. Internet Architecture

Conceptually, the Internet can be viewed as a collection of domains, each of which consists of a network or a relatively small internet operated by a single organization such as a university, company, or government agency. Each domain is an autonomous system that can be configured as the local authority desires. It might consist of a single computer or a complex internet consisting of many LANs, MANs, and even WANs.

oversee /ˌəʊvə'si:/v. 监视; 检查;
管理**nonprofit**

/nɒn'prɒfɪt/

a. 非营利的

registrar

/ˌredʒɪ'strɑː, 'redʒɪs-/

n. 域名注册服务商

The establishment of domains is **overseen** by the Internet Corporation for Assigned Names and Numbers (ICANN²), which is a **nonprofit** corporation established to coordinate the naming of domains and the assignment of Internet addresses. To establish a domain on the Internet, the domain must first be registered via one of the companies, called **registrars**, that have been delegated for this purpose by ICANN.³

Once a domain has been registered, it can be attached to the existing Internet by means of a router that connects one of the networks in the domain to a network already in the Internet. This particular router is often referred to as the domain's gateway, in that⁴ it represents the domain's gate to the rest of the Internet. From the point of view of a single domain, the portion of the Internet lying outside of its gateway is sometimes called the cloud, in reference to the fact that the structure of the Internet outside the domain's gateway is out of the domain's control and is of minimal concern to the operation of the domain itself.⁵ Any message being transmitted to a

¹ **DARPA**: (美国) 国防部高级研究计划局 (Defense Advanced Research Projects Agency 的首字母缩略)。

² **ICANN**: 互联网名称与数字地址分配机构 (Internet Corporation for Assigned Names and Numbers 的首字母缩略), 读作 /'aɪkæn/, 成立于 1998 年, 总部设在美国加利福尼亚州洛杉矶县的海里纳德雷 (Marina del Rey)。

³ *the companies, called registrars, that have been delegated for this purpose by ICANN*: 得到 ICANN 授权做此工作、被称为域名注册服务商的公司。

⁴ *in that*: 因为, 原因在于。

⁵ *From the point of view of a single domain, the portion of the Internet lying outside of its gateway is sometimes called the cloud, in reference to the fact that the structure of the Internet outside the domain's gateway is out of the domain's control and is of minimal concern to the operation of the domain itself.*: 从单个域的角度看, 位于其网关外面的那部分因特网有时被称为“云”, 这是指以下的事实: 其网关外面的因特网结构不受其控制, 而且与其本身的操作也没有什么关系。

destination within the domain is handled within the domain; any message being transmitted to a destination outside the domain is directed toward the gateway where it is sent out into the cloud.

If one “stood” at a domain’s gateway and “looked out” into the cloud, one could find a variety of structures. Indeed, the Internet has grown in a somewhat **haphazard** way as various domains have found points at which to connect to the cloud. One popular structure, however, is for the gateways of a number of domains to be connected to form a regional network of gateways. For instance, a group of universities could choose to **pool** their resources to construct such a network. In turn, this regional network would be connected to a more global network to which other regional networks attach.¹ In this manner, that portion of the cloud takes on a hierarchical structure (Figure 8A-1).

haphazard

/ˌhæpˈhæzəd/

a. 无计划的, 随意的

pool /pu:l/

v. 把...集中一起用, 共用

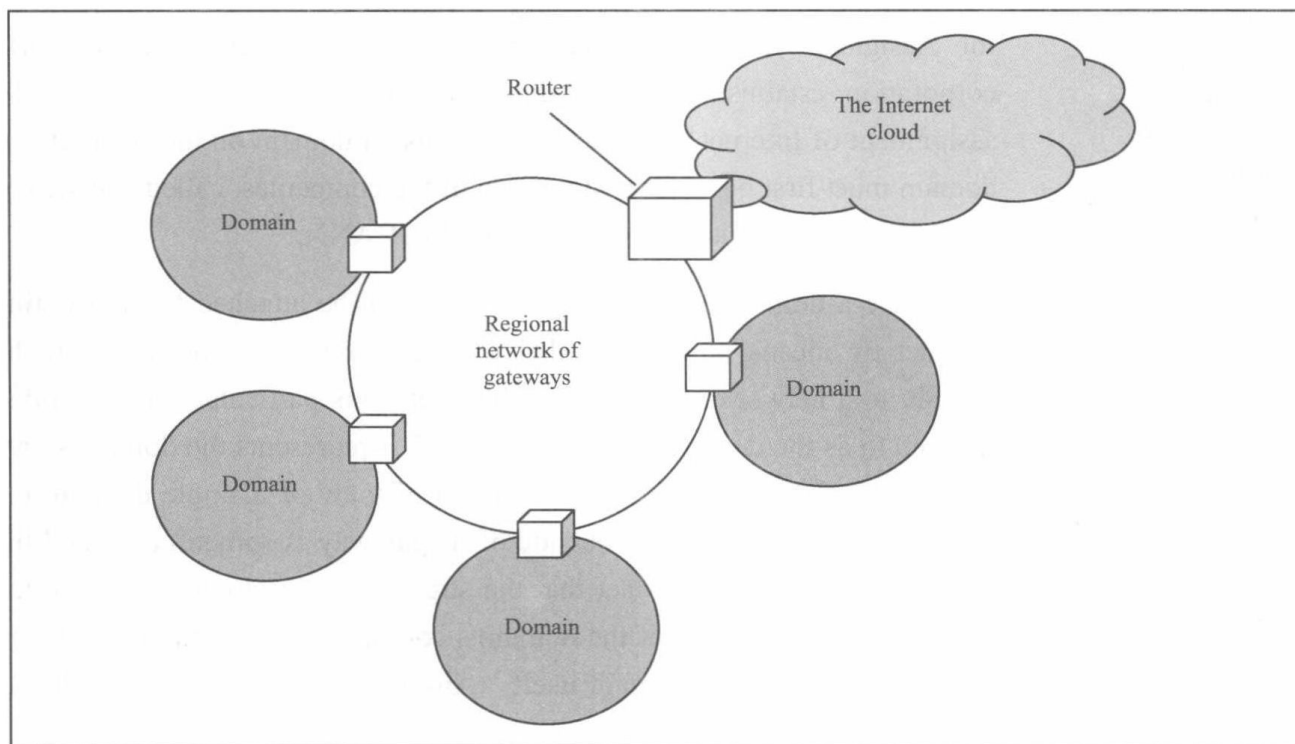


Figure 8A-1: A Typical Approach to Connecting to the Internet

II. Connecting to the Internet

To simplify the process of connecting to the Internet, numerous

¹ For instance, a group of universities could choose to pool their resources to construct such a network. In turn, this regional network would be connected to a more global network to which other regional networks attach.: 例如, 一组大学可以选择将其资源集中起来, 用于构建这样一个网络。而这个区域网络接着又连接到一个更大的网络上, 后者上面还连接有其他的区域网络。global network 本来表示“全局(或全球)网络”。在本句中, a more global network 是相对于 a regional network 而言, 是指比其更大的网络。

dial-up /'daɪəlʌp/

a. 拨号的

arena /ə'ri:nə/

n. 竞技场;

活动场所

cable television

有线电视

broadband

/'brɔ:dbænd/

a. 宽带的

transmitter

/trænz'mɪtə; træns-/

n. 发射机; 发送器

hotspot /'hɒtspɒt/

n. (网络) 热点,
热区

cyber /'saɪbə/

a. 计算机(网络)的

cyber cafe

网吧

plaza /'plɑ:zə, 'plæz-/

n. (城市建筑群中间
或附近的) 购物区;
(城市中的) 广场

companies, called Internet service providers (ISPs), allow customers to connect their domains to the Internet via the ISP's equipment or to become a part of a domain already established by the ISP. Perhaps the least expensive connections to an ISP are obtained through temporary telephone links called **dial-up** connections. Using this approach, an individual connects his or her computer to the local telephone line and executes a software package that places a call to a computer at the ISP. At this point the ISP provides Internet access for the duration of the telephone call.

This temporary connection model is popular (and, in fact, expanding) in the cell phone/hand-held computer **arena**. However, in fixed location applications it is rapidly giving way to permanent connections through higher bandwidth systems such as DSL¹, **cable television** lines, and satellite links. These technologies have greatly reduced the expense of **broadband** (high-capacity) Internet access (which used to be available only to large organizations that could afford the expense of dedicated Internet connections), making today's multimedia applications accessible to home users.

Another development taking place at the household level is that home computer systems are expanding from single computer installations to multicomputer networks, usually implemented as Ethernet bus networks. These networks gain access to the Internet by means of a router within the home that is permanently connected to an ISP via DSL, a cable television line, or a satellite link, making the home system a local network within a domain established by the ISP. The connection from the home's router to the local computers is often wireless. That is, each computer communicates with the router by means of a small **transmitter**/receiver with a range of approximately 100 feet. Such wireless local networks are popular outside the home as well where they are used to create **hotspots** in which any wireless-enabled computer (laptop or hand held) that comes within range can access the Internet.² Examples include office buildings, "**cyber cafes**," public **plazas**, and entire cities.

¹ **DSL**: 数字用户线路 (Digital Subscriber Line 的首字母缩略)。

² *Such wireless local networks are popular outside the home as well where they are used to create hotspots in which any wireless-enabled computer (laptop or hand held) that comes within range can access the Internet.*: 这种无线局部网在家庭之外也很流行, 被用于建立热区。在这些热区的范围内, 任何具有无线上网能力的计算机(膝上型或手持式计算机)均可访问因特网。本句有3个定语从句: 由 where 引导的定语从句比较特殊, 其先行词(即修饰对象)实际上是 outside the home, 这种用法不多见; 由 in which 引导的定语从句的先行词为 hotspots; 由 that 引导的定语从句的先行词为 computer。后两个定语从句依次镶嵌在前一个定语从句中。

In closing we should note that ISPs provide much more than mere access to the Internet. In many cases they are the companies that construct and maintain the Internet infrastructure. Moreover, their services generally include a wide array of both hardware and software support for their customers. In fact, an ISP might provide a significant portion of the computing resources (such as data storage and data processing services) required by a client.

addressing

/ə'dresɪŋ/

n. 编址; 寻址

III. Internet Addressing

An internet must be associated with an internet-wide addressing system that assigns an identifying address to each computer in the system. In the Internet these addresses are known as IP¹ addresses. Each IP address is a pattern of 32 bits, although plans are currently underway to increase this to 128 bits. Each 32-bit address consists of two parts: a pattern identifying the domain in which the computer resides and a pattern identifying the particular computer within the domain. The part of the address identifying the domain, the **network identifier**, is assigned under the authority of ICANN at the time the domain is registered. Thus it is by means of this registration process that each domain in the Internet is assured of having a unique network identifier. The portion of the address that identifies a particular computer within a domain is called the **host address**. The host address is assigned by the domain's local authority—usually a person having a job title such as network administrator or system administrator.

identifier

/aɪ'dentɪfaɪə/

n. 标识符

network identifier

网络标识符

host /həʊst/

n. 主机

host address

主机地址

decimal notation

十进制记数法

dotted decimal notation

点分十进制记数法

base ten notation

以 10 为底的记数法

binary notation

二进制记数法

binary /'baɪnəri/

n. 二进制 (数)

IP addresses are traditionally written in **dotted decimal notation** in which the bytes of the address are separated by periods and each byte is expressed as an integer represented in traditional **base ten notation**.² For example, using dotted decimal notation, the pattern 5.2 would represent the two-byte bit pattern 0000010100000010, which consists of the byte 00000101 (represented by 5) followed by the byte 00000010 (represented by 2), and the pattern 17.12.25 would represent the three-byte bit pattern consisting of the byte 00010001 (which is 17 written in **binary notation**), followed by the byte 00001100 (12 written in **binary**), followed by the byte 00011001 (25 written in **binary**). Thus a computer in the domain of the Addison-Wesley³ publishing company might have the IP address

¹ IP: IP 协议, 网际协议, 网间协议 (Internet Protocol 的首字母缩略)。

² the bytes of the address are separated by periods and each byte is expressed as an integer represented in traditional base ten notation: 地址的各字节用圆点分隔, 每个字节用一个整数来表示, 而该整数是用传统的以 10 为底的记数法来表示的。

³ Addison-Wesley: 艾迪生-韦斯利出版公司, 系培生 (Pearson) 教育出版集团旗下的著名出版公司, 主要出版计算机、经济、金融、数学、统计学等方面的图书及电子读物。

192.207.177.133, where the first three bytes (192.207.177) form the network identifier (identifying the Addison-Wesley domain) and the last byte (133) is the host address (identifying a particular computer within Addison-Wesley's domain).

conductive

/kən'dju:siv/

a. 有助的,
有益的 (*to*)

mnemonic

/ni:'monik/

a. 助记的; 记忆的

mnemonic address

助记地址

domain name

域名

suffix / 'sʌfiks /

n. 后缀

top-level domain

顶级域名

country code

国家代码

mnemonic name

助记名, 缩写名

Addresses in bit-pattern form (even when compressed using dotted decimal notation) are rarely **conductive** to human consumption.¹ For this reason each domain is also assigned a unique **mnemonic address** known as a **domain name**. For example, the domain name of the Addison-Wesley publishing company is `aw.com`. Note that the naming system reflects the domain's classification, which in this case is commercial as indicated by the `com` **suffix**. Such a classification is called a **top-level domain**² (TLD). There are numerous TLDs, including `edu` for educational institutions, `gov` for U.S. government institutions, `org` for nonprofit organizations, `museum` for museums, `info` for unrestricted use, and `net`, which was originally intended for Internet service providers but is now used on a much broader scale. In addition to these general TLDs, there are also two-letter TLDs for specific countries (called **country-code** TLDs) such as `au` for Australia and `ca` for Canada.

Once a domain has a **mnemonic name**, its local authority is free to extend it to obtain mnemonic names for the machines within its domain. For example, an individual machine within the domain `aw.com` might be identified as `ssenterprise.aw.com`.

We should emphasize that the dotted notation used in mnemonic addresses is not related to the dotted decimal notation used to represent IP addresses. Instead, the sections in a mnemonic address identify the computer's location within a hierarchical classification system. In particular, the address `ssenterprise.aw.com` indicates that the computer known as `ssenterprise` is within the institution `aw` within the class (or TLD) of commercial domains `com`. In the case of large domains, a local authority might break its domain into subdomains, in which case the mnemonic addresses of the computers within the domain might be longer. For example, suppose Nowhere University³ was assigned the domain name `nowhereu.edu` and chose to divide its domain into subdomains. Then, a computer at

¹ *Addresses in bit-pattern form (even when compressed using dotted decimal notation) are rarely conducive to human consumption.*: 用位模式形式表示的地址 (即使经过使用点分十进制记数法压缩) 很少能够有助于人们理解记忆。

² *top-level domain*: 有时亦称为 top-level domain name (TLDN)。

³ *Nowhere University*: 乌有大学, 虚构的大学。

Nowhere University could have an address such as r2d2.compsc.nowhereu.edu, meaning that the computer r2d2 is in the subdomain compsc within the domain nowhereu within the class of educational domains edu.

name server
名字服务器

domain name system
域名系统

Each domain's local authority is responsible for maintaining a directory containing the mnemonic address and the corresponding IP address of those computers within its domain. This directory is implemented on a designated computer within the domain in the form of a server, called a **name server**, that responds to requests regarding address information. Together, all the name servers throughout the Internet are part of an Internet-wide directory system known as the **domain name system** (DNS) that is used to convert addresses in mnemonic form into their equivalent bit-pattern forms. In particular, when a human requests that a message be sent to a destination identified in mnemonic form, DNS is used to convert that mnemonic address into its equivalent IP address that is compatible with the Internet software. The process of extracting information from DNS is often referred to as a "DNS lookup." Normally, a DNS lookup is completed in a fraction of a second.

Exercises

I. Fill in the blanks with the information given in the text:

1. The Internet originated from _____ projects going back to the early 1960s.
2. The establishment of domains on the Internet is overseen by a nonprofit corporation called _____.
3. A registered domain can be attached to the existing Internet by means of a(n) _____ which is often referred to as the _____ of the domain.
4. A user typically connects to the Internet in one of two ways: _____ connections and _____ connections.
5. A(n) _____ is a company that provides Internet access and other services to customers.
6. An IP address is a 32-bit address made up of two parts: the

_____ identifier and the _____ address, identifying the domain in which a computer resides and the particular computer within the domain respectively.

7. IP addresses are traditionally written in dotted _____ notation.

8. For ease of memorization, each domain is also assigned a unique _____ address known as a domain name.

II. Translate the following terms or phrases from English into Chinese and vice versa:

- | | |
|-----------------------------------|--------------|
| 1. cell phone | 11. 助记地址 |
| 2. IP address | 12. 网吧 |
| 3. autonomous system | 13. 宽带因特网访问 |
| 4. dial-up connection | 14. 顶级域名 |
| 5. network identifier | 15. 因特网编址 |
| 6. binary notation | 16. 点分十进制记数法 |
| 7. mnemonic name | 17. 因特网服务提供商 |
| 8. Internet-wide directory system | 18. 专用因特网连接 |
| 9. name server | 19. 主机地址 |
| 10. Internet infrastructure | 20. 硬件与软件支持 |

III. Fill in each of the blanks with one of the words given in the following list, making changes if necessary:

- | | | | |
|----------------|------------------|----------------------|-------------------|
| <i>origin</i> | <i>risk</i> | <i>channel</i> | <i>connection</i> |
| <i>sender</i> | <i>Internet</i> | <i>own</i> | <i>academic</i> |
| <i>network</i> | <i>computer</i> | <i>technological</i> | <i>data</i> |
| <i>develop</i> | <i>establish</i> | <i>researcher</i> | <i>packet</i> |

Early computer networks used leased telephone company lines for their connections. Telephone company systems of that time established a single connection between _____ and receiver for each telephone call, and that connection carried all _____ along a single path. When a company wanted to connect computers it _____ at two different locations, the company placed a telephone call to _____ the connection, and then connected one computer to each end of that single _____.

The U.S. Defense Department was concerned about the inherent

_____ of this single-channel method for connecting computers, and its researchers _____ a different method of sending information through multiple _____. In this method, files and messages are broken into _____ that are labeled electronically with codes for their _____, sequences, and destinations. In 1969, Defense Department _____ in the Advanced Research Projects Agency (ARPA) used this network model to connect four _____ into a network called the ARPANET. The ARPANET was the earliest of the _____ that eventually combined to become what we now call the _____. Throughout the 1970s and 1980s, many researchers in the _____ community connected to the ARPANET and contributed to the _____ developments that increased its speed and efficiency.

IV. Translate the following passage from English into Chinese:

The Internet provides only the physical and logical infrastructure that connects millions of computers together. Many believe that the World Wide Web (WWW, or simply the Web) provides the *killer application* (制胜法宝) for this global network. The Web is considered the content of the Internet, providing all sorts of information by using a rich set of tools that manage and link text, graphics, sound, and video. Providing and viewing information on the Web is accomplished using server applications and client applications.

If you've already explored the Web, you'll recognize the client-side application as the Web browser. A Web browser receives, interprets, and displays pages of information from the Web. The user can navigate within pages, jump to other pages by clicking hypertext links, and point to just about any page on the Web.

Section B

The Layered Approach to Internet Software

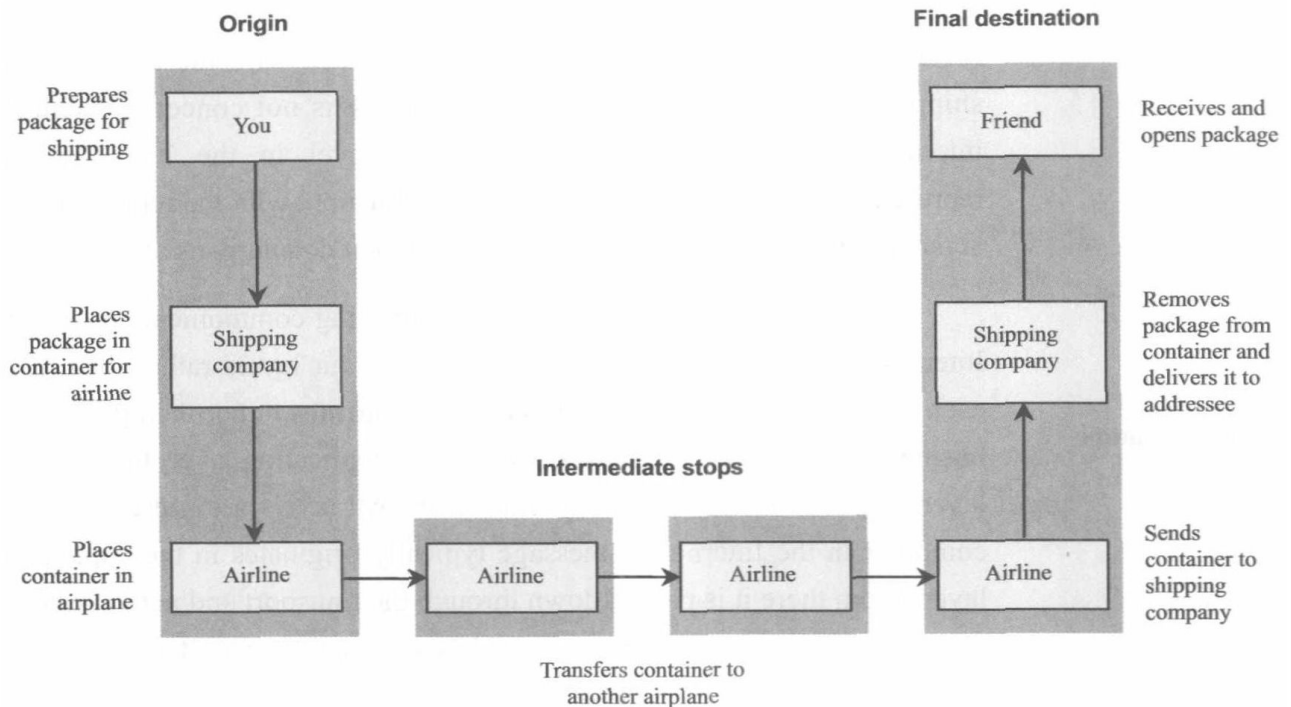


Figure 8B-1: Package-Shipping Example

A principal task of networking software is to provide the infrastructure required for transferring messages from one machine to another. In the Internet, this message-passing activity is accomplished by means of a hierarchy of software units, which perform tasks analogous to those that would be performed if you were to send a gift in a package from the West Coast of the United States to a friend on the East Coast (Figure 8B-1). You would first wrap the gift as a package and write the appropriate address on the outside of the package. Then, you would take the package to a shipping company such as the U.S. Postal Service¹. The shipping company might place the package along with others in a large container and deliver the container to an airline, whose services it has contracted.² The airline would place the container in an aircraft and transfer it to the destination city,

analogous

/ə'neɪləɡəs/
a. 相似的;
可比拟的 (to/with)

shipping /'ʃɪpɪŋ/
n. 运送, 运输;
航运业

¹ the U.S. Postal Service: 美国邮政管理局。

² deliver the container to an airline, whose services it has contracted: 将集装箱送往与其签有服务合同的航空公司。

addressee

/ˌædre'si:/

n. 收信人; 收件人

perhaps with intermediate stops along the way. At the final destination, the airline would remove the container from the aircraft and give it to the shipping company's office at the destination. In turn, the shipping company would take your package out of the container and deliver it to the **addressee**.

In short, the transportation of the gift would be carried out by a three-level hierarchy: (1) the user level (consisting of you and your friend), (2) the shipping company, and (3) the airline. Each level uses the next lower level as an abstract tool. (You are not concerned with the details of the shipping company, and the shipping company is not concerned with the internal operations of the airline.) Each level in the hierarchy has representatives at both the origin and the destination, with the representatives at the destination tending to do the reverse of their counterparts at the origin.¹

software routine

软件例程

Such is the case with software for controlling communication over the Internet, except that the Internet software has four layers rather than three, each consisting of a collection of **software routines** rather than people and businesses. The four layers are known as the application layer, the transport layer, the network layer, and the link layer. All layers are present on each computer in the Internet. A message typically originates in the application layer. From there it is passed down through the transport and network layers as it is prepared for transmission, and finally it is transmitted by the link layer. The message is received by the link layer at the destination and passed back up the hierarchy² until it is delivered to the application layer at the message's destination.

Let us investigate this process more thoroughly by tracing a message as it finds its way through the system (Figure 8B-2). We begin our journey with the application layer.

The application layer consists of those software units such as clients and servers that use Internet communication to carry out their tasks.³ Although the names are similar,⁴ this layer is not restricted to software in

¹ *Each level in the hierarchy has representatives at both the origin and the destination, with the representatives at the destination tending to do the reverse of their counterparts at the origin.*: 分层结构中的每一层在起始地点和目的地均有代表, 在目的地的代表所做的工作往往与在起始地点的代表所做的工作相反。

² *passed back up the hierarchy*: 在分层结构中向上回传。

³ *those software units such as clients and servers that use Internet communication to carry out their tasks*: 客户机软件和服务器软件等使用因特网通信来完成其任务的软件单元。本句中的 clients 和 servers 指客户机与服务器上安装的相关软件。

⁴ *the names are similar*: 这里所说的“名称相似”是指 application layer (应用层) 与 application software (应用软件) 都包含 application。

login /'login; 'lɔ:g-/
n. 注册, 登录, 进入系统

telnet /'telnet/
n. [亦作 **Telnet**]
远程登录协议 (或程序、服务)

utility software
实用软件

the application classification, but also includes many utility packages. For example, software for transferring files using FTP¹ or for providing remote login capabilities using telnet have become so common that they are normally considered utility software.

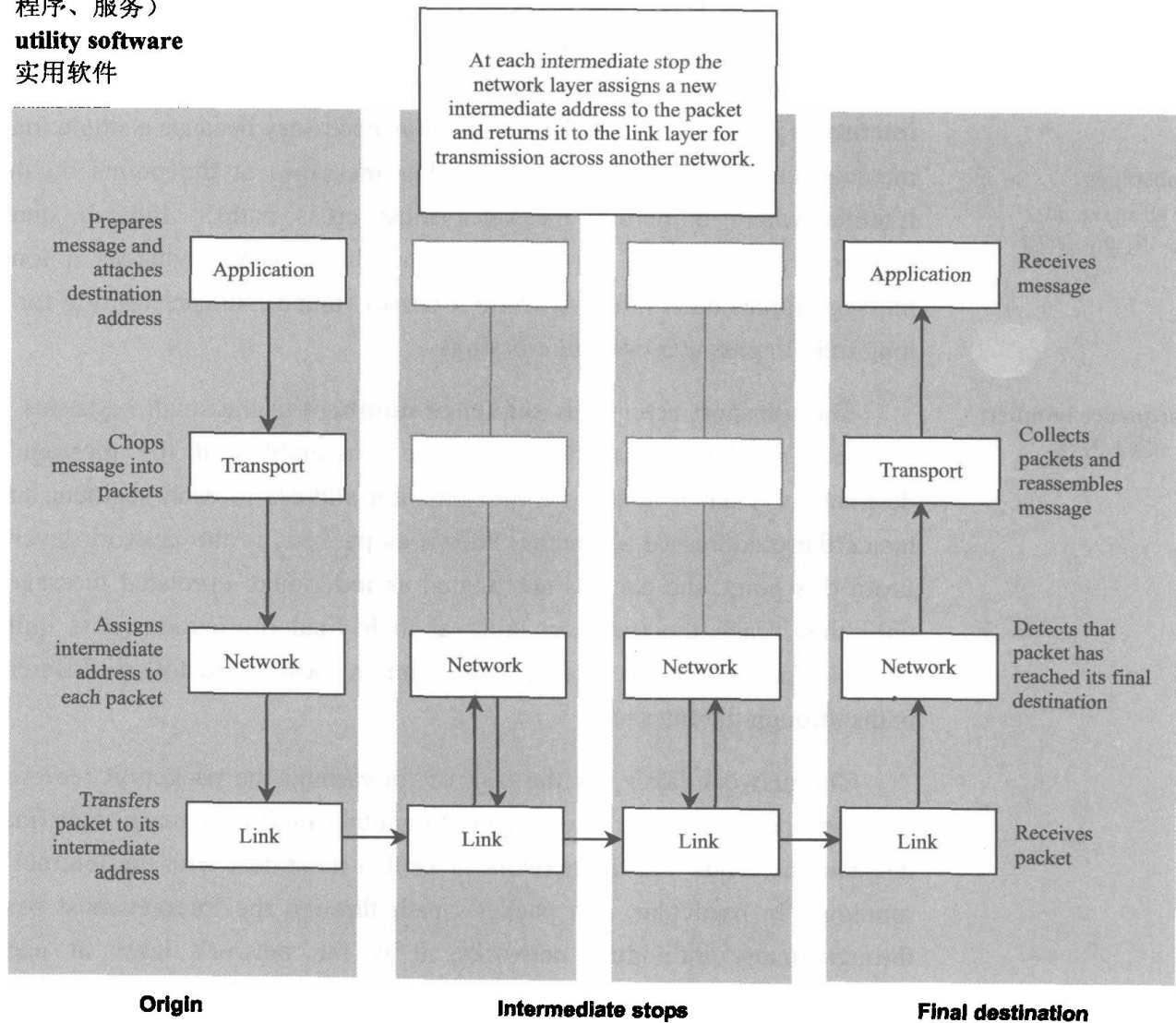


Figure 8B-2: Following a Message through the Internet

The application layer uses the transport layer to send and receive messages over the Internet in much the same way that you would use a shipping company to send and receive packages. Just as it is your responsibility to provide an address compatible with the specifications of the shipping company, it is the application layer's responsibility to provide an address that is compatible with the transport layer. (To fulfill this need,

¹ **FTP**: 文件传送协议 (File Transfer Protocol 的首字母缩略)。

the application layer may use the services of the name servers within the Internet to translate mnemonic addresses used by humans into Internet-compatible IP addresses.)

obstruct

/əb'strakt, ɒb-/

v. 阻塞; 阻碍

A major task of the transport layer is to accept messages from the application layer and to ensure that the messages are properly formatted for transmission over the Internet. Toward this latter goal, the transport layer divides long messages into small segments, which are transmitted over the Internet as individual units. This division is necessary because a single long message can obstruct the flow of other messages at the points on the Internet where numerous messages must cross paths.¹ Indeed, small segments of messages can interweave at these points, whereas a long message forces others to wait while it passes (much like cars waiting for a long train to pass at a railroad crossing).

sequence number

(顺) 序号

The transport layer adds sequence numbers to the small segments it produces so that the segments can be reassembled at the message's destination. Then it attaches the destination address to each segment and hands these addressed segments, known as packets, to the network layer.² From this point, the packets are treated as individual, unrelated messages until they reach the transport layer at their final destination. It is quite possible for the packets related to a common message to follow different paths through the Internet.

The network layer has the task of forwarding the packets it receives from one network within the Internet to another until they reach their final destinations. Thus it is the network layer that must deal with the Internet's topology. In particular, if a packet's path through the Internet must pass through many individual networks, it is the network layer at each intermediate stop that determines the direction in which the packet will be sent next. The decision is made as follows: If the final destination of the packet is within the current network, the network layer will send the packet there; otherwise, the network layer will send the packet to a router in the current network through which the packet can be transferred into an adjacent network. In this way, a packet destined for a computer within the current network will be sent to that machine, whereas a packet destined for a computer outside the current network will continue its network-to-

destine /'destin/

v. 预定, 指定 (往某一目的地或供某种用途) (*for*)

¹ *numerous messages must cross paths*: 许多报文必须路径相交。

² *hands these addressed segments, known as packets, to the network layer*: 把这些编好地址、称为分组的片段交给网络层。

network journey.

append / ə'pend /
v. 附加

Having determined a packet's next destination, the network layer **appends** this address to the packet as an intermediate address and hands the packet to the link layer.

The link layer has the responsibility of transferring the packet to the intermediate address that was determined by the network layer. Thus the link layer must deal with the communication details particular to the individual network in which the computer resides. If that network is a token ring,¹ the link layer must wait for possession of the token before transmitting. If the network uses CSMA/CD, the link layer must listen for a silent bus before transmitting.

coincide
/ ,kəuɪn'saɪd /
v. 相符, 相一致;
重合
hop / hɒp /
v. 跳跃

When a packet is transmitted, it is received by the link layer at the computer designated by the local address attached to the message. There, the link layer hands the packet up to its network layer where the packet's final destination is compared to the current location. If these do not **coincide**, the network layer determines a new intermediate address for the packet, attaches that address to the packet, and returns the packet to the link layer for retransmission. In this manner, each packet **hops** from machine to machine on its way to its final destination. Note that only the link and network layers are involved at the intermediate stops during this journey (see again Figure 8B-2).

If the network layer determines that an arriving packet has reached its final destination, it hands the packet to its transport layer. As the transport layer receives packets from the network layer, it extracts the underlying message segments and reconstructs the original message according to the sequence numbers that were provided by the transport layer at the message's origin. Once the message is assembled, the transport layer hands it to the appropriate unit within the application layer—thus completing the message transmission process.

incoming
/ 'ɪn,kʌmɪŋ /
a. 进来的; 输入的
port / pɔ:t /
n. 端口, 通信口
port number
端口号

Determining which unit within the application layer should receive an **incoming** message is an important task of the transport layer. This is handled by assigning unique **port numbers** to the various units and requiring that the appropriate port number be appended to a message's address before starting the message on its journey. Then, once the message

¹ *that network is a token ring*: 该网络是权标环网络。token ring 意为“权标环”“令牌环”，但在本句中指采用权标环协议进行通信的网络。

is received by the transport layer at the destination, the transport layer merely hands the message to the application layer software at the designated port number.

Users of the Internet rarely need to be concerned with port numbers because the common applications have universally accepted port numbers. For example, if a Web browser is asked to retrieve the document whose URL¹ is `http://www.zoo.org/animals/frog.html`, the browser assumes that it should contact the HTTP server at `www.zoo.org` via port number 80. Likewise, when transferring a file, an FTP client assumes that it should communicate with the FTP server through port numbers 20 and 21.

In summary, communication over the Internet involves the interaction of four layers of software. The application layer deals with messages from the application's point of view. The transport layer converts these messages into packets that are compatible with the Internet and reassembles messages that are received before delivering them to the appropriate application. The network layer deals with directing the packets through the Internet. The link layer handles the actual transmission of packets from one machine to another. With all this activity, it is somewhat amazing that the response time of the Internet is measured in milliseconds, so that many transactions appear to take place instantaneously.

Exercises

I. Fill in the blanks with the information given in the text:

1. For an outgoing message, the four layers of software involved in Internet communication work in the order of the _____ layer, the _____ layer, the _____ layer, and the _____ layer; for an incoming message, it is just the opposite.

2. The _____ layer converts long messages into packets compatible with the Internet before their transmission and reassembles them at the destination.

3. The packets related to a common message may follow _____ paths through the Internet.

¹ URL: 统一资源定位符, 统一资源定位器 (Uniform Resource Locator 的首字母缩略), 亦称为 Universal Resource Locator。

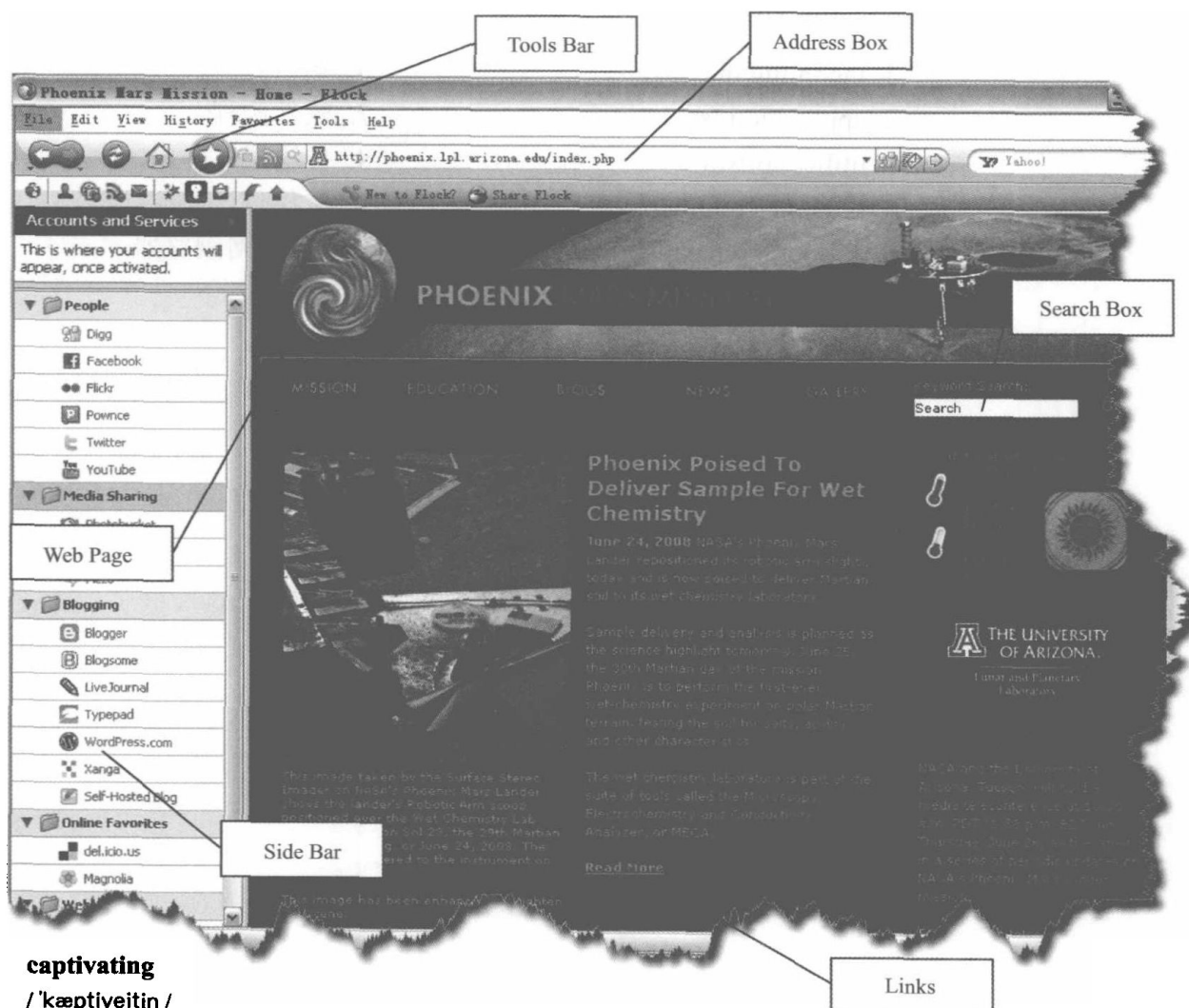
4. Only the _____ and _____ layers are involved at the intermediate stops during a packet's journey to its final destination.

II. Translate the following terms or phrases from English into Chinese and vice versa:

- | | |
|------------------------------|------------|
| 1. incoming message | 6. 端口号 |
| 2. application layer | 7. 软件例程 |
| 3. utility software | 8. 传输层 |
| 4. sequence number | 9. 文件传送协议 |
| 5. remote login capabilities | 10. 万维网浏览器 |

Section C

Web Basics



captivating

/ˈkæptɪveɪtɪŋ/

a. 迷人的, 可爱的

instrumental

/ˌɪnstrə'mentəl/

.instrə-/

a. 起作用的, 有帮助的 (*in, to*)

ferry /ˈferi/

v. 渡运; 运送

animation

/ˌæni'meɪʃən/

n. 动画 (制作)

hyperlink

/ˈhaɪpəlɪŋk/

n. 超 (级) 链接

I. The World Wide Web

One of the Internet's most **captivating** attractions, the World Wide Web (also referred to as WWW, W3, or the Web) is a collection of files that can be linked and accessed using HTTP. HTTP (Hypertext Transfer Protocol) is the communications standard that's **instrumental** in **ferrying** Web documents to all corners of the Internet.

Many Web-based files produce documents called Web pages. Other files contain photos, videos, **animations**, and sounds that can be incorporated into specific Web pages. Most Web pages contain links (sometimes called "**hyperlinks**") to related documents and media files.

cyberspace

/ˈsaɪbəspeɪs/
n. 计算机空间,
网络空间

jewelry /ˈdʒuːəlri/

n. 〈美〉[总称] 珠宝, 首饰

host /haʊst/

v. 作...的主机

Web server

万维网服务器

locator

/ləʊˈkeɪtə; ˈləukeɪtə/
n. 定位器, 定位符

Uniform Resource Locator

统一资源定位符,
统一资源定位器

main page

主页

doorway /ˈdɔːweɪ/

n. 出入口, 门口

home page

主页

desk /desk/

n. 服务台; 部门

showbiz /ˈʃəʊbiz/

n. 〈口〉娱乐性
行业; 娱乐圈
(= show business)

hurricane

/ˈhʌrɪkən/
n. 飓风

A series of Web pages can be grouped into a Web site—a sort of virtual “place” in **cyberspace**. Every day, thousands of people shop at Nordstrom’s¹ Web site, an online department store featuring clothing, shoes, and **jewelry**.² Thousands of people visit the Webopedia³ Web site to look up the meaning of computer terms. At the ABC⁴ News Web site, people not only read about the latest news, sports, and weather, but also discuss current issues with other readers. The Web encompasses these and hundreds of thousands of other sites.

Web sites are **hosted** by corporate, government, college, and private computers all over the world. The computers and software that store and distribute Web pages are called **Web servers**.

Every Web page has a unique address called a URL (**Uniform Resource Locator**). For example, the URL for the Cable News Network⁵ Web site is *http://www.cnn.com*. Most URLs begin with *http://* to indicate the Web’s standard communications protocol. When typing a URL, the *http://* can usually be omitted, so *www.cnn.com* works just as well as *http://www.cnn.com*.

Most Web sites have a **main page** that acts as a “doorway” to the rest of the pages at the site. This main page is sometimes referred to as a “**home page**,” although this term has another meaning.⁶ The URL for a Web site’s main page is typically short and to the point, like *www.cnn.com*.

The pages for a Web site are typically stored in topic area folders, which are reflected in the URL. For example, the CNN site might include a weather center at *www.cnn.com/weather/* and an entertainment **desk** at *www.cnn.com/showbiz/*. A series of Web pages are then grouped under the appropriate topic. For example, you might find a page about **hurricanes** at the URL *www.cnn.com/weather/hurricanes.html*, and you could find a page about El Niño⁷ at *www.cnn.com/weather/elnino.html*. The file name of a

¹ *Nordstrom*: 诺茨罗姆公司, 美国服饰零售商, 总部设在西雅图。

² *an online department store featuring clothing, shoes, and jewelry*: 一家主要销售衣服、鞋子和珠宝的在线百货公司。

³ *Webopedia*: 网络百科全书, 与计算机和互联网技术有关的在线词典和搜索引擎。

⁴ *ABC*: 美国广播公司 (American Broadcasting Company 的首字母缩略)。

⁵ *Cable News Network*: (美国) 有线电视新闻网, 首字母缩略词为 CNN。

⁶ *home page*: 既可指一家网站的主页 (网站的首页), 也可指为浏览器设的主页 (每次打开浏览器时所显示的网页)。

⁷ *El Niño*: 厄尔尼诺现象。El Niño 来自西班牙语, 意为“圣婴”, 即上帝之子。厄尔尼诺现象的基本特征是, 赤道太平洋中东部海域大范围海水温度异常升高, 海水水位上涨。该现象发生时, 往往出现全球范围的灾害天气。

Hypertext Markup Language

超(级)文本标记语言

space / speɪs /

n. 空格, 空白

punctuation mark

标点符号

underline

/ 'ʌndəˌlaɪn /

n. 下划线

forward slash

正斜杠

duplicate

/ 'dʒuːplikeɪt /

v. 复制; 重复

capitalization

/ ,kæpɪtəlaɪ'zeɪʃən;

-li'z /

n. 大写字母的使用

lowercase

/ 'ləʊə'keɪs /

a. (字母) 小写的

cosmetic

/ kɒz'metɪk /

a. 化妆用的; 装饰性的; 非实质性的

boxer / 'bɒksə /

n. 拳击运动员, 拳师

Address box

地址框

Enter key

回车键

HTML tag

HTML 标记

specific Web page always appears last in the URL—*hurricanes.html* and *elnino.html* are the names of two Web pages. Web page file names usually have an .htm or .html extension, indicating that the page was created with HTML (Hypertext Markup Language), a standard format for Web documents.

A URL never contains a **space**, even after a **punctuation mark**, so do not type any spaces within a URL. An **underline** character is sometimes used to give the appearance of a space between words, as in the URL *www.detroit.com/top_10.html*. Be sure to use the correct type of slash—always a **forward slash** (/)—and **duplicate** the URL's **capitalization** exactly. The servers that run some Web sites are case sensitive,¹ which means that an uppercase letter is not the same as a lowercase letter. On these servers, typing *www.cmu.edu/Overview.html* (with an uppercase “O”) will not locate the page that's actually stored as *www.cmu.edu/overview.html* (with a lowercase “o”).

II. Browsers

A Web browser—usually referred to simply as a browser—is a software program that runs on your computer and helps you access Web pages. Two of today's most popular browsers are Microsoft Internet Explorer² (IE) and Netscape Navigator (Navigator). A browser provides a sort of “window” in which it displays a Web page. The borders of the window contain a set of menus and controls to help you navigate from one Web page to another. Despite small **cosmetic** differences and some variations in terminology, Web browsers offer a remarkably similar set of features and capabilities.

A browser fetches and displays Web pages. Suppose that you want to view the Web page located at *www.e-course.com/boxer.html*. You enter the URL into a special **Address box** that's provided by your browser. When you press the **Enter key**, the browser contacts the Web server at *www.e-course.com* and requests the *boxer.html* page. The server sends your computer the data stored in *boxer.html*. This data includes two things: the information you want to view, and embedded codes, called **HTML tags**, that tell your browser how to display it. The tags specify details such as the

¹ *The servers that run some Web sites are case sensitive*: 运行有些网站的服务器区分大小写。这里的 case 指“(字母的)大小写”, sensitive 表示“敏感的”, case sensitive 意指“对大小写敏感的”, 即“区分大小写的”。

² *Microsoft Internet Explorer*: 微软因特网浏览器, 美国微软(Microsoft)公司开发的网络浏览器。

background color, the text color and size, and the placement of graphics.

Copy command

复制命令

Most browsers provide a **Copy command** that allows you to copy a section of text from a Web page, which you can then paste into one of your own documents. To keep track of the source for each text section, you can use the Copy command to record the Web page's URL from the Address box, and then paste the URL into your document.

III. Search Engines

The term search engine popularly refers to a Web site that provides a variety of tools to help you find information. Search engines such as Google¹ and Yahoo!² are indispensable tools when it comes to finding information on the Web. Depending on the search engine you use, you can find information by entering a description, filling out a form, or clicking a series of links to drill down through a list of topics and subtopics. Based on your input, the search engine displays a list of Web pages like the one shown in Figure 8C-1.

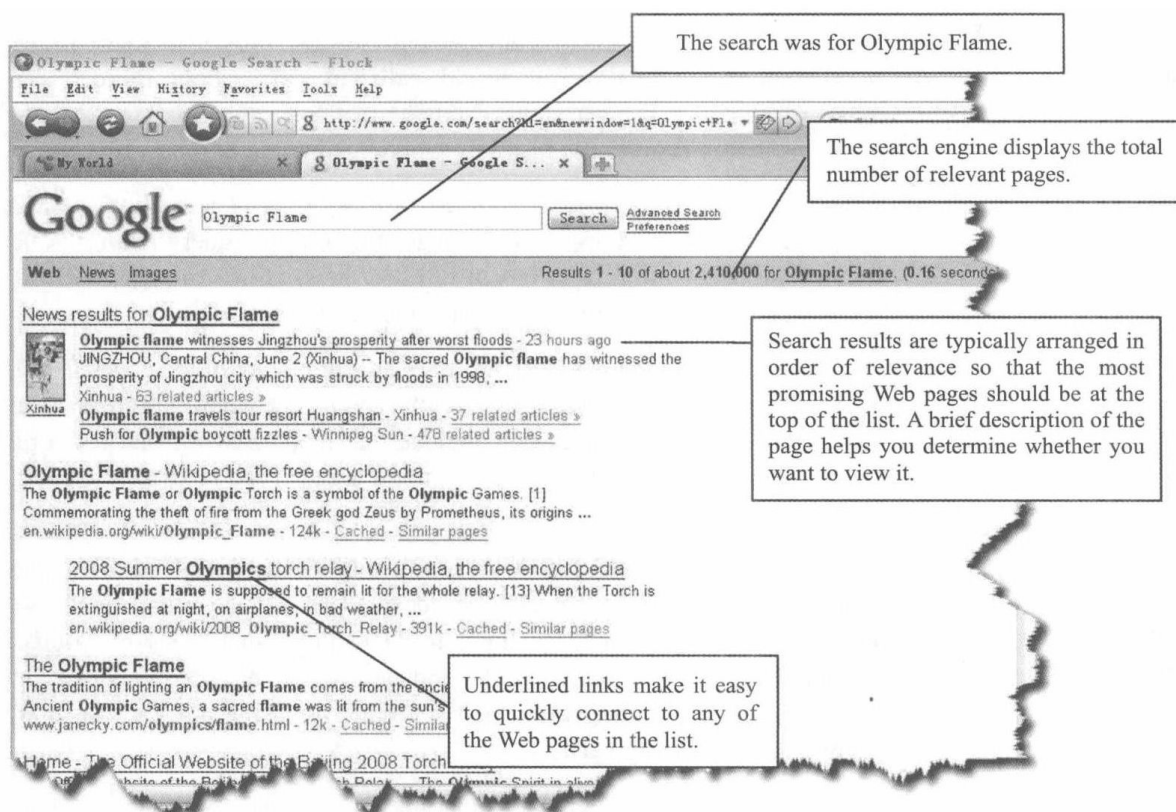


Figure 8C-1: In response to a query, a search engine produces a list of relevant Web pages, along with a brief description of each page and a link to it.

¹ **Google:** 谷歌, 网上搜索引擎。同名公司成立于1998年, 总部设在美国加利福尼亚州芒廷维尤 (Mountain View)。

² **Yahoo!:** 雅虎, 网上搜索引擎。同名公司成立于1994年, 总部设在美国加利福尼亚州森尼韦尔 (Sunnyvale)。

keyword /'ki:wə:d/
n. 关键词, 关键字
operator /'ɒpəreɪtə/
n. (运)算子, 算符

inundate /'ɪnəndeɪt/
v. 淹没; (似洪水般) 布满
formulate
/'fɔ:mjuleɪt/
v. 构想出; 系统地阐述

railroad car
(美) 火车车厢

ferry /'feri/
n. 渡船; 摆渡;
渡口

quotation marks
[复] (一对) 引号
immigration
/ˌɪmɪ'ɡreɪʃən/
n. 移民, 移居
green /ɡri:n/
n. (高尔夫)球穴区;
高尔夫球场
greeting card
贺卡

A query describes the information you want to find. It includes one or more **keywords** and can also include **search operators**. A keyword (sometimes called a “search term”) is any word that describes the information you’re trying to find. You can enter more than one search term. Separate each term with a space or a search operator.

Search engines have a tendency to **inundate** you with possibilities—often finding thousands of potentially relevant Web pages. To receive a more manageable list of results, you need to **formulate** a more specific search. A search operator is a word or symbol that describes a relationship between keywords and thereby helps you create a more focused query. The search operators that you can use with each search engine vary slightly. To discover exactly how to formulate a query for a particular search engine, refer to its Help pages. Most search engines allow you to formulate queries with the search operators described in Figure 8C-2.

AND	When two search terms are joined by AND, both terms must appear on a Web page before it can be included in the search results. The query <i>railroad AND cars</i> will locate pages that contain both the words “railroad” and “cars.” Your search results might include pages containing information about old railroad cars , about railroad car construction, and even about railroads that haul automobiles (“cars”). Some search engines use the plus symbol (+) instead of the word AND.
OR	When two search terms are joined by OR, either one or both of the search words could appear on a page. Entering the query <i>railroad OR cars</i> produces information about railroad fares, railroad routes, railroad cars, automobile safety records, and even car ferries .
NOT	The keyword following NOT must not appear on any of the pages found by the search engine. Entering <i>railroad NOT cars</i> would tell the search engine to look for pages that include “railroad” but not the keyword “cars.” In some search engines, the minus sign (–) can be used instead of the word NOT.
Quotation Marks	Surrounding a series of keywords with quotation marks indicates that the search engine must treat the words as a phrase. The complete phrase must exist on a Web page for it to be included in the list of results. Entering “ <i>green card</i> ” would indicate that you are looking for information on immigration , not information on the color green, golf greens, or greeting cards.
NEAR	The NEAR operator tells a search engine that you want documents in which one of the keywords is located close to but not necessarily next to the other keyword. The query <i>library NEAR/15 congress</i> means that the words “library” and “congress” must appear within 15 words of each other. Successful searches could include documents containing phrases such as “Library of Congress” or “Congress funds special library

wildcard

/ˈwaɪldkɑːd/

n. 通配符, 万能符

wildcard character

通配符

derivation

/ˌderɪˈveɪʃən/

n. 派生(物)

medic /ˈmedɪk/

n. 〈口〉(随军急救)

卫生队队员; 医科

学生; 医生

medication

/ˌmediˈkeɪʃən/

n. 药物治疗; 药物

medicinal

/meˈdɪsɪnəl/

a. (医)药的; 药用的;

有疗效的

backcountry

/ˈbæk.kʌntri/

a. 偏僻乡村的

recipe /ˈresɪpi/

n. 烹饪法, 食谱;

处方

colon /ˈkəʊlən/

n. 冒号

targeted /ˈtɑːɡɪtɪd/

a. 有目标的; 指向

目标的; 被定为目

标的

	research.”
Wildcards	The asterisk (*) is sometimes referred to as a “wildcard character.” It allows a search engine to find pages with any derivation of a basic word. For example, the query <i>medic*</i> would not only produce pages containing the word “medic,” but also “medical,” “medication,” “medicine,” and “medicinal.”
Field Searches	Some search engines allow you to search for a Web page by its title or by any part of its URL. The query <i>T:Backcountry Recipe Book</i> indicates that you want to find a specific Web page titled “Backcountry Recipe Book.” In this search, the <i>T:</i> tells the search engine to look at Web page titles, and the information following the colon identifies the name of the title.

Figure 8C-2: Search Operators

A topic directory is a list of topics and subtopics, such as Arts, Business, Computers, and so on, which are arranged in a hierarchy (Figure 8C-3). The top level of the hierarchy contains general topics. Each successive level of the hierarchy contains increasingly specific subtopics.¹ A topic directory might also be referred to as a “category list,” an “index,” or a “directory.” To use a topic directory, simply click a general topic. When a list of subtopics appears, click the one that’s most relevant to the information you are trying to locate. If your selection results in another list of subtopics, continue to select the most relevant one until the search engine presents a list of Web pages. You can then link to these pages just as though you had used a keyword query.

Many search engines provide an advanced search form that helps you formulate a very **targeted** search. A search form helps you enter complex queries. It might also allow you to search for pages that are written in a particular language, located on a specific Web server, or created within a limited range of dates. It is usually accessible by clicking an Advanced Search link, which often is located on the main page of the search engine Web site.

Instead of entering a cryptic query such as *movie+review+“The Producers”*, wouldn’t it be nice to enter a more straightforward question like *Where can I find a review of The Producers?* A few search engines specialize in natural language queries, which accept questions written in plain English.

¹ *Each successive level of the hierarchy contains increasingly specific subtopics.*: 分层结构顶层下面依次相继的各层含有越来越具体的子主题。

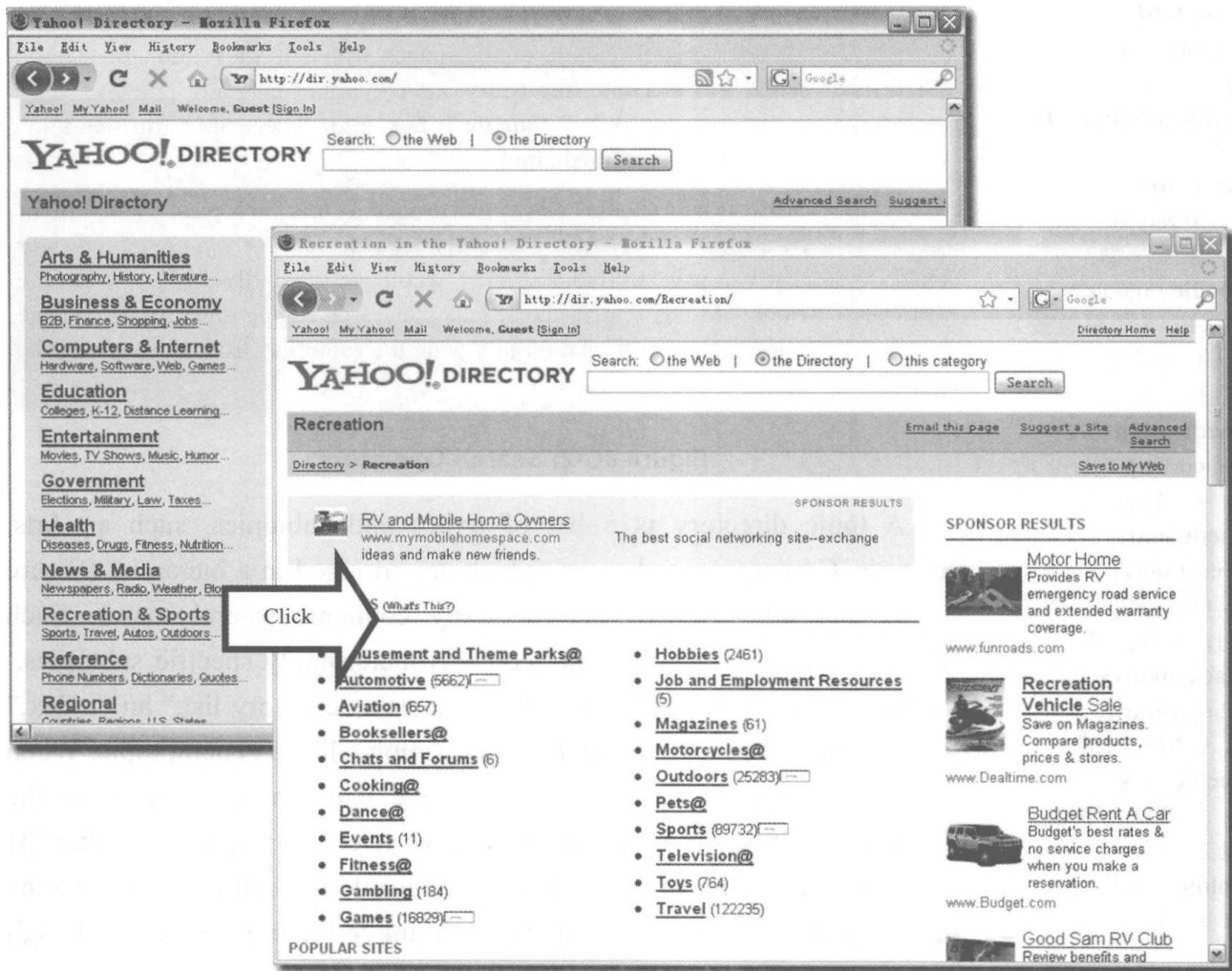


Figure 8C-3: To use a topic directory, simply click a general topic. When a list of subtopics appears, click the one that's most relevant to the information you are trying to locate.

Exercises

I. Fill in the blanks with the information given in the text:

1. Most Web pages contain links, sometimes called _____, to related documents and media files.
2. The main page of a Web site is sometimes called its _____ page.
3. HTML _____ are sets of instructions inserted into an HTML document to provide formatting and display information to a Web browser.

4. When you use a search engine, a query includes one or more _____ and can also include search _____.

II. Translate the following terms or phrases from English into Chinese and vice versa:

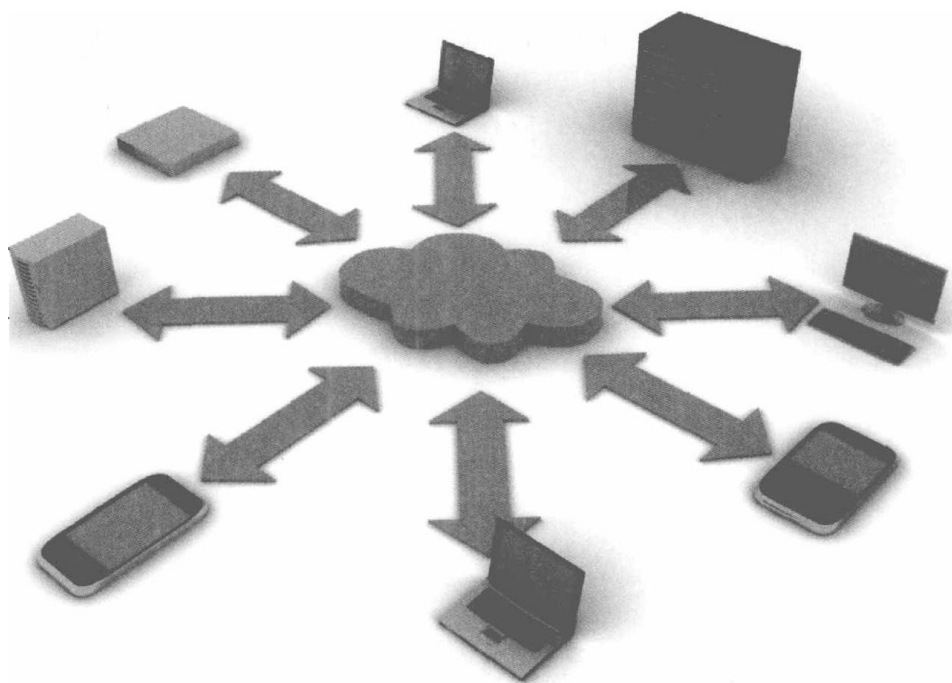
- | | |
|-----------------------|------------|
| 1. wildcard character | 5. 回车键 |
| 2. Copy command | 6. 搜索引擎 |
| 3. search operator | 7. 嵌入代码 |
| 4. home page | 8. 超文本标记语言 |

Unit 9 Mobile and Cloud Computing

(移动与云计算)

Section A

Cloud Computing



virtualize

/ˈvɜ:tʃuəlaɪz/

v. 虚拟化

workload

/ˈwɜ:klaʊd/

n. 工作量; 工作负荷

batch /bætʃ/

n. 批, 批量, 成批

backend /ˈbækend/

a. 后端的

grid /grid/

n. 网格, 格网

leverage

/ˈli:vərɪdʒ; ˈle-/

v. 充分利用

I. Introduction

Cloud computing has been defined differently by many users and designers. IBM, a major player in cloud computing, has defined it as follows: “A cloud is a pool of virtualized computer resources. A cloud can host a variety of different workloads, including batch-style backend jobs and interactive and user-facing applications.”¹

The concept of cloud computing has evolved from cluster, grid, and utility computing.² Cluster and grid computing leverage the use of many

¹ “A cloud is a pool of virtualized computer resources. A cloud can host a variety of different workloads, including batch-style backend jobs and interactive and user-facing applications.”: 一个云是一个虚拟化计算机资源池。云可托管各种不同的工作负荷, 包括批处理的后端作业和交互式面向用户应用。

² The concept of cloud computing has evolved from cluster, grid, and utility computing.: 云计算这个概念从集群计算、网格计算和效用计算发展而来。

throughput

/'θru:put /

n. 吞吐量

whereby /'hwɛə'baɪ /

ad. 靠那个; 借以

server farm

大型机服务器

outsource /aut'sɔ:s /

v. (将...) 外包

scenario /si'nɑ:riəu /

n. 方案; 情况;

脚本

desktop /'deskɒp /

n. 台式(计算)机;

桌面

utilization

/ju:tilaɪ'zeɪʃən; -li'z- /

n. 利用

virtualization

/vɜ:tʃuəlaɪ'zeɪʃən /

n. 虚拟化

elastic /i'læstɪk /

a. 有弹性的; 可伸缩的

provision

/prə'vɪʒən /

v. 供应, 提供

server cluster

服务器集群

hybrid /'haɪbrɪd /

a. 混合的

subscription

/səb'skrɪpʃən /

n. 订购; 订阅

callout /'kɔ:laut /

n. 标注

callout box

标注框

computers in parallel to solve problems of any size. Utility and Software as a Service (SaaS¹) provide computing resources as a service with the notion of pay per use. Cloud computing leverages dynamic resources to deliver large numbers of services to end users. Cloud computing is a high-throughput computing (HTC²) paradigm whereby the infrastructure provides the services through a large data center or server farms. The cloud computing model enables users to share access to resources from anywhere at any time through their connected devices.

The cloud will free users to focus on user application development and create business value by outsourcing job execution to cloud providers. In this scenario, the computations (programs) are sent to where the data is located, rather than copying the data to millions of desktops as in the traditional approach. Cloud computing avoids large data movement, resulting in much better network bandwidth utilization. Furthermore, machine virtualization has enhanced resource utilization, increased application flexibility, and reduced the total cost of using virtualized data-center resources.

The cloud offers significant benefit to IT³ companies by freeing them from the low-level task of setting up the hardware (servers) and managing the system software. Cloud computing applies a virtual platform with elastic resources put together by on-demand provisioning of hardware, software, and data sets, dynamically. The main idea is to move desktop computing to a service-oriented platform using server clusters and huge databases at data centers. Cloud computing leverages its low cost and simplicity to benefit both providers and users. Cloud computing intends to leverage multitasking to achieve higher throughput by serving many heterogeneous applications, large or small, simultaneously.

II. Public, Private, and Hybrid Clouds

A public cloud is built over the Internet and can be accessed by any user who has paid for the service. Public clouds are owned by service providers and are accessible through a subscription. The callout box in top of Figure 9A-1 shows the architecture of a typical public cloud. Many public clouds are available, including Google App Engine (GAE)⁴, Amazon

¹ SaaS: 软件即服务 (Software as a Service 的首字母缩略)。

² HTC: 高吞吐(量)计算 (high-throughput computing 的缩略)。

³ IT: 信息技术 (information technology 的首字母缩略)。

⁴ Google App Engine: 谷歌应用引擎, 一种平台即服务云计算平台, 可让用户在 Google 的基础架构上运行其网络应用程序。

aforementioned

/ə'fɔ:menʃənd/

a. 前面提到的

Web Services (AWS)¹, Microsoft Azure², IBM Blue Cloud³, and Salesforce.com's Force.com⁴. The providers of the aforementioned clouds are commercial providers that offer a publicly accessible remote interface for creating and managing VM⁵ instances within their proprietary infrastructure. A public cloud delivers a selected set of business processes. The application and infrastructure services are offered on a flexible price-per-use basis.

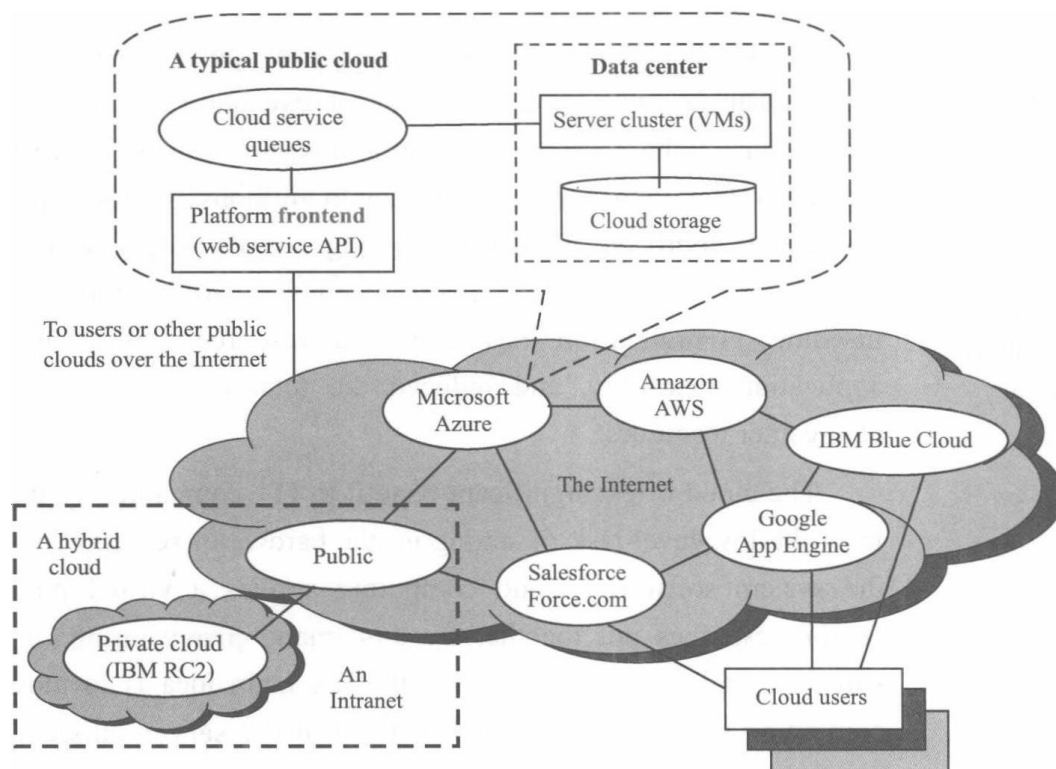


Figure 9A-1: Public, private, and hybrid clouds illustrated by functional architecture and connectivity of representative clouds

frontend

/ˈfrʌntˈend/

n. 前端

deployment

/diˈplɔɪmənt/

n. 部署; 展开

A *private cloud* is built within the domain of an intranet owned by a single organization. Therefore, it is client owned and managed, and its access is limited to the owning clients and their partners. Its **deployment** was not meant to sell capacity over the Internet through publicly accessible

¹ *Amazon Web Services*: 亚马逊网络服务, 亚马逊公司提供的云计算服务。亚马逊公司成立于1994年, 1995年开始在线经营, 总部设在美国华盛顿州西雅图市。

² *Microsoft Azure*: 即 Windows Azure, 微软提供的一个开放而灵活的云平台, 用户通过该平台可在微软管理的数据中心的全球网络中快速生成、部署和管理应用程序。

³ *IBM Blue Cloud*: 国际商用机器公司 (IBM) 的“蓝云”云计算解决方案。

⁴ *Salesforce.com's Force.com*: 企业云计算公司 Salesforce.com 的 Force.com 网站, 该网站拓展了该公司在企业云计算领域的领导地位。Salesforce 是创建于1999年的一家客户关系管理 (CRM) 软件服务提供商, 总部设在美国加利福尼亚州旧金山。Salesforce 译作“软件营销部队”或“软营”。

⁵ *VM*: 虚拟机 (virtual machine 的首字母缩略)。

agile /'ædʒaɪl/

a. 敏捷的; 灵活的

customization

/kʌstəmaɪ'zeɪʃən/

n. 定制, 用户化

resiliency

/ri'zɪliənsi/

n. 弹性; 复原力

deskside /'desksaɪd/

a. 桌边(型)的

pay-as-you-go

/'peɪəzju'geu/

n. & a. 即付即用
(的)

interfaces. Private clouds give local users a flexible and **agile** private infrastructure to run service workloads within their administrative domains. A private cloud is supposed to deliver more efficient and convenient cloud services. It may impact the cloud standardization, while retaining greater **customization** and organizational control. Intranet-based private clouds are linked to public clouds to get additional resources.

A *hybrid cloud* is built with both public and private clouds, as shown at the lower-left corner of Figure 9A-1. Private clouds can also support a hybrid cloud model by supplementing local infrastructure with computing capacity from an external public cloud. For example, the Research Compute Cloud (RC2) is a private cloud, built by IBM, that interconnects the computing and IT resources at eight IBM Research Centers scattered throughout the United States, Europe, and Asia. A hybrid cloud provides access to clients, the partner network, and third parties.

In summary, public clouds promote standardization, preserve capital investment, and offer application flexibility. Private clouds attempt to achieve customization and offer higher efficiency, **resiliency**, security, and privacy. Hybrid clouds operate in the middle, with many compromises in terms of resource sharing.

III. Cloud Design Objectives

Despite the controversy surrounding the replacement of desktop or **deskside** computing by centralized computing and storage services at data centers or big IT companies, the cloud computing community has reached some consensus on what has to be done to make cloud computing universally acceptable. The following list highlights six design objectives for cloud computing:

- **Shifting computing from desktops to data centers:** Computer processing, storage, and software delivery is shifted away from desktops and local servers and toward data centers over the Internet.
- **Service provisioning and cloud economics:** Providers supply cloud services by signing service-level agreements (SLAs¹) with consumers and end users. The services must be efficient in terms of computing, storage, and power consumption. Pricing is based on a **pay-as-you-go** policy.
- **Scalability in performance:** The cloud platforms and software and

¹ SLA: 服务水平(或级别、等级)协议(service-level agreement 的缩略)。

scale /skeɪl/

v. 攀登; (阶梯) 逐步升高

interoperable

/ˌɪntəˈɒpərəbl/

a. 可互操作的, 能互用的

lock-in /ˈlɒkɪn/

n. 锁定; 同步

infrastructure services must be able to **scale** in performance as the number of users increases.

- **Data privacy protection:** Can you trust data centers to handle your private data and records? This concern must be addressed to make clouds successful as trusted services.
- **High quality of cloud services:** The Quality of Service (QoS) of cloud computing must be standardized to make clouds **interoperable** among multiple providers.
- **New standards and interfaces:** This refers to solving the data **lock-in** problem associated with data centers or cloud providers. Universally accepted APIs and access protocols are needed to provide high portability and flexibility of virtualized applications.

IV. Enabling Technologies for Clouds

Clouds are enabled by the progress in hardware, software, and networking technologies summarized in Table 9A-1.

Technology	Requirements and Benefits
Fast platform deployment	Fast, efficient, and flexible deployment of cloud resources to provide dynamic computing environment to users
Virtual clusters on demand	Virtualized cluster of VMs provisioned to satisfy user demand and virtual cluster reconfigured as workload changes
Multitenant techniques	SaaS for distributing software to a large number of users for their simultaneous use and resource sharing if so desired
Massive data processing	Internet search and web services which often require massive data processing, especially to support personalized services
Web-scale communication	Support for e-commerce, distance education, telemedicine , social networking , digital government, and digital entertainment applications
Distributed storage	Large-scale storage of personal records and public archive information which demands distributed storage over the clouds
Licensing and billing services	License management and billing services which greatly benefit all types of cloud services in utility computing

**Table 9A-1: Cloud-Enabling Technologies
in Hardware, Software, and Networking**

These technologies play instrumental roles in making cloud computing a reality. Most of these technologies are mature today to meet increasing demand. In the hardware area, the rapid progress in multi-core CPUs, memory chips, and **disk arrays** has made it possible to build faster data centers with huge amounts of storage space. Resource virtualization enables

multitenant

/ˌmʌltiˈtenənt/

a. 多租户的

personalize

/ˈpɜːsənəlaɪz/

v. 使个性化

telemedicine

/ˈteliˌmedɪsɪn/

n. 远程医学

social networking

社交网络

archive /ˈɑːkaɪv/

n. [常作~s] 档案;

档案馆, 档案室

licensing

/ˈlaɪsənsɪŋ/

n. 发给许可证; 许可, 特许

billing /ˈbɪlɪŋ/

n. 开(账)单; 记账

disk array

磁盘阵列

rapid cloud deployment and disaster recovery. Service-oriented architecture (SOA) also plays a vital role.

Progress in providing SaaS, Web 2.0 standards, and Internet performance have all contributed to the emergence of cloud services. Today's clouds are designed to serve a large number of **tenants** over massive volumes of data. The availability of large-scale, distributed storage systems is the foundation of today's data centers. Of course, cloud computing is greatly benefitted by the progress made in license management and automatic billing techniques in recent years.

V. A Generic Cloud Architecture

Figure 9A-2 shows a security-aware cloud architecture.¹ The Internet cloud is **envisioned** as a massive cluster of servers. These servers are provisioned on demand to perform collective web services or distributed applications using data-center resources. The cloud platform is formed dynamically by provisioning or deprovisioning servers, software, and database resources. Servers in the cloud can be physical machines or VMs. User interfaces are applied to request services. The provisioning tool **carves** out the cloud system to deliver the requested service.²

In addition to building the server cluster, the cloud platform demands distributed storage and accompanying services. The cloud computing resources are built into the data centers, which are typically owned and operated by a third-party provider. Consumers do not need to know the underlying technologies. In a cloud, software becomes a service. The cloud demands a high degree of trust of massive amounts of data retrieved from large data centers. We need to build a framework to process large-scale data stored in the storage system. This demands a distributed file system over the database system. Other cloud resources are added into a cloud platform, including storage area networks (SANs³), database systems, firewalls, and security devices. Web service providers offer special APIs that enable developers to exploit Internet clouds. Monitoring and metering units⁴ are used to track the usage and performance of provisioned resources.

tenant / 'tenənt /
n. 租户, 承租人

envision / in'vi:ʒən /
v. 想象; 设想

carve / kɑ:v /
v. 雕刻, 刻

¹ *Figure 9A-2 shows a security-aware cloud architecture.*: 图 9A-2 显示了一种具有安全意识的云体系结构。

² *The provisioning tool carves out the cloud system to deliver the requested service.*: 服务提供工具塑造出云系统, 来提供所请求的服务。carve out 意为“(雕)刻出”, “开辟出”, “创(业)”。

³ *SAN*: 存储区域网(络)(storage area network 的首字母缩略)。

⁴ *monitoring and metering units*: 指“监控与计量设备”。

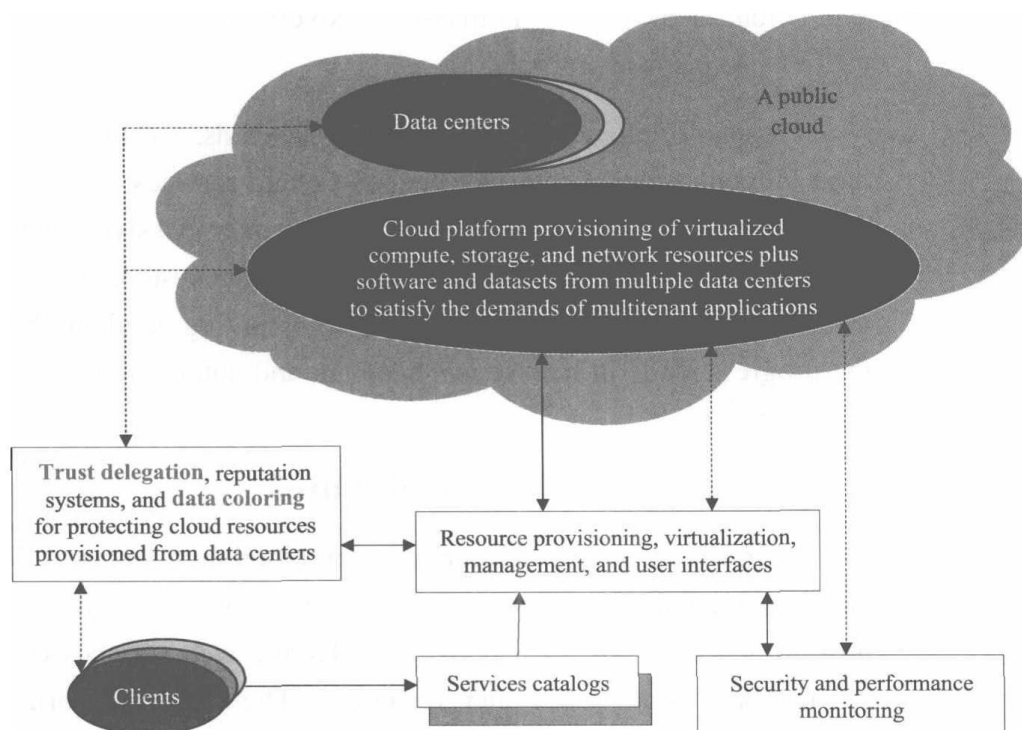


Figure 9A-2: A security-aware cloud platform built with a virtual cluster of VMs, storage, and networking resources over the data-center servers operated by providers

trust delegation
信任委托

data coloring
数据着色

hydroelectric
/ˌhaɪdrəʊ'lektrɪk/
a. 水力发电的;
(电) 由水力发的

safeguard
/'seɪfɡɑːd/
v. 保护, 维护

The software infrastructure of a cloud platform must handle all resource management and do most of the maintenance automatically. Software must detect the status of each node server joining and leaving, and perform relevant tasks accordingly. Cloud computing providers, such as Google and Microsoft, have built a large number of data centers all over the world. Each data center may have thousands of servers. The location of the data center is chosen to reduce power and cooling costs. Thus, the data centers are often built around **hydroelectric** power. The cloud physical platform builder is more concerned about the performance/price ratio and reliability issues than sheer speed performance.

In general, private clouds are easier to manage, and public clouds are easier to access. The trends in cloud development are that more and more clouds will be hybrid. This is because many cloud applications must go beyond the boundary of an intranet. One must learn how to create a private cloud and how to interact with public clouds in the open Internet. Security becomes a critical issue in **safeguarding** the operation of all cloud types.

Exercises

I. Fill in the blanks with the information given in the text:

1. The concept of cloud computing has evolved from _____, grid, and _____ computing.
2. A public cloud is built over the _____ and can be accessed by any user who has paid for the service.
3. A private cloud is built within the domain of a(n) _____ owned by a single organization.
4. A(n) _____ cloud is built with both public and private clouds.
5. Generally speaking, _____ clouds are easier to manage, and _____ clouds are easier to access.
6. The rapid progress in _____ CPUs, memory chips, and _____ arrays has made it possible to build faster data centers with huge amounts of storage space.
7. One of the design objectives for cloud computing is shifting computing from _____ to data centers.
8. The Internet cloud is envisioned as a massive cluster of servers, which can be _____ machines or VMs.

II. Translate the following terms or phrases from English into Chinese and vice versa:

- | | |
|--------------------------------------|---------------|
| 1. server farm | 11. 云计算提供商 |
| 2. access protocol | 12. 存储芯片 |
| 3. storage area network | 13. 基于内部网的私有云 |
| 4. high-throughput computing | 14. 网络带宽 |
| 5. server cluster | 15. 混合云 |
| 6. public cloud | 16. 磁盘阵列 |
| 7. grid computing | 17. 软件即服务 |
| 8. security-aware cloud architecture | 18. 集群计算 |
| 9. social networking | 19. 虚拟化计算机资源 |
| 10. utility computing | 20. 多核处理器 |

III. Fill in each of the blanks with one of the words given in the following list, making changes if necessary:

<i>service</i>	<i>cloud</i>	<i>marketing</i>	<i>storage</i>
<i>automated</i>	<i>layer</i>	<i>resource</i>	<i>management</i>
<i>software</i>	<i>result</i>	<i>develop</i>	<i>platform</i>
<i>application</i>	<i>SaaS</i>	<i>foundation</i>	<i>serve</i>

The architecture of a cloud is developed at three layers: infrastructure, platform, and application. The infrastructure _____ is deployed first to support IaaS (Infrastructure as a Service) services. This layer serves as the _____ for building the platform layer of the cloud for supporting PaaS (Platform as a Service) _____. In turn, the platform layer is a foundation for implementing the _____ layer for SaaS applications.

The infrastructure layer is built with virtualized compute, _____, and network resources. The abstraction of these hardware _____ is meant to provide the flexibility demanded by users. Internally, virtualization realizes _____ provisioning of resources and optimizes the infrastructure management process. The _____ layer is for general-purpose and repeated usage of the collection of _____ resources. This layer provides users with an environment to _____ their applications, to test operation flows, and to monitor execution _____ and performance. In a way, the virtualized cloud platform _____ as a “system *middleware* (中间件)” between the infrastructure and application layers of the _____. The application layer is formed with a collection of all needed software modules for _____ applications. Service applications in this layer include daily office _____ work. The application layer is also heavily used by enterprises in business _____ and sales, consumer relationship management, financial transactions, and supply chain management.

IV. Translate the following passage from English into Chinese:

Cloud computing applies a virtualized platform with elastic resources on demand by provisioning hardware, software, and data sets dynamically. The idea is to move desktop computing to a service-oriented platform using server clusters and huge databases at data centers. Cloud computing leverages its low cost and simplicity to benefit both users and providers. Machine virtualization has enabled such cost-effectiveness. Cloud

computing intends to satisfy many user applications simultaneously. The cloud *ecosystem* (生态系统) must be designed to be secure, trustworthy, and dependable. Some computer users think of the cloud as a centralized resource pool. Others consider the cloud to be a server cluster which practices distributed computing over all the servers used.

Traditionally, a distributed computing system tends to be owned and operated by an autonomous administrative domain (e.g., a research laboratory or company) for *on-premises* (机构内部的) computing needs. However, these traditional systems have encountered several performance *bottlenecks* (瓶颈): constant system maintenance, poor utilization, and increasing costs associated with hardware/software *upgrades* (升级). Cloud computing as an on-demand computing paradigm resolves or relieves us from these problems.

command line
命令行

tablet computer
平板电脑
smart card
智能卡

e-label /'i:leibəl/
n. 电子标签

synergistic
/.sɪnə'dʒɪstɪk/
a. 增效的; 协同作
用的; 协作的

merchandise
/'mɜ:tʃəndaɪz, -daɪs/
n. [总称] 商品

modulate
/'mɒdjuleɪt/
v. 调制

and user interfaces. Contemporary human-computer interaction models, whether **command-line**, menu-driven, or GUI-based, are inappropriate and inadequate to meet ubiquitous computing demands. The natural IoT paradigm appropriate to a ubiquitous computing world has yet to emerge. Contemporary devices that lend support to ubiquitous computing include smartphones, **tablet computers**, sensor networks, RFID tags, **smart cards**, GPS devices, and others.

In ubiquitous computing, the IoT provides a network of sensor- or radio-connected devices that can be uniquely identified and located in the cyber-physical space. This IoT is mostly wirelessly connected as a self-configuring network of radio-frequency tags, low-cost sensors, or **e-labels**. The term “IoT” combines RFID technology with today’s IPv6-based Internet technology¹. All things (objects) have IP addresses, which can be uniquely identified. The IP-identifiable objects are readable, recognizable, locatable, addressable, and/or controllable via the Internet, aided by RFID, Wi-Fi², ZigBee³, mobile networks, and GPS.⁴

II. Enabling and Synergistic Technologies

Many technologies can be applied to build the IoT infrastructure and specific IoT systems for special application domains. Supportive technologies are divided into two categories. Enabling technologies build up the foundations of the IoT. Among the enabling technologies, tracking (RFID), sensor networks, and GPS are critical.

RFID is applied with electronic labels or RFID tags on any objects being monitored or tracked. The tags may be applied to any objects, such as **merchandise**, tools, smartphones, computers, animals, or people. The purpose is to identify and track the objects using radio waves or sensing signals. Some tags can be read from tens or hundreds of meters away via a wireless reader. Most RFID tags contain at least two major parts. One is an integrated circuit for storing and processing information, **modulating** and

¹ *IPv6-based Internet technology*: 基于 IPv6 的因特网技术。IPv6 是 Internet Protocol version 6 的缩略, 指 IP 协议第 6 版。

² *Wi-Fi*: *Wireless Fidelity* 的缩略, 字面意思为“无线保真”。Wi-Fi 是一种无线传输规范。Wi-Fi 网络使用 IEEE 802.11b 或 802.11a 无线电技术, 提供安全、可靠、快速的无线连通性。

³ *ZigBee*: 基于 IEEE 802.15.4 标准的低功耗个域网协议。其特点是近距离、低复杂度、自组织、低功耗、低数据速率、低成本。ZigBee 这一名称来源于蜜蜂 (bee) 通过锯齿形舞蹈 (zigzag) 向同伴传递蜜源方位、距离等信息的行为, 也就是说蜜蜂依靠这样的方式构成了群体中的通信网络。

⁴ *The IP-identifiable objects are readable, recognizable, locatable, addressable, and/or controllable via the Internet, aided by RFID, Wi-Fi, ZigBee, mobile networks, and GPS.*: 在射频识别、Wi-Fi、ZigBee、移动网络及 GPS 的帮助下, 可用 IP 地址标识的对象能够通过因特网来读取、识别、定位、寻址和/或控制。

demodulate

/di:'mɒdjuleɪt/

v. 解调

antenna /æn'tenə/

n. 天线

spatial /'speɪʃəl/

a. 空间的

cooperative

/kəu'ɒpəreɪtɪv/

a. 合作的, 协作的

pollutant

/pə'lʊ:tənt/

n. 污染性物质

battlefield

/'bætlfi:ld/

n. 战场

habitat /'hæbɪtæt/

n. (动植物的) 自然生存环境, 栖息地

navigation

/ˌnævi'geɪʃən/

n. 导航; 航行

geotagging

/ˌdʒi:əu'tæɡɪŋ/

n. 地理标记, 地理标签

geocaching

/ˌdʒi:əu'kæʃɪŋ/

n. 地理藏宝

biometrics

/ˌbaɪəu'metɪks/

n. 生物统计学

machine vision

机器视觉

augment /ɔ:g'ment/

v. 扩大; 增加

augmented reality

增强现实

energy harvesting

能量收集

telepresence

/'teli,pri:zəns/

n. 远程呈现

autonomy

/ɔ:'tɒnəmi/

n. 自治(权), 自主性

demodulating a radio-frequency (RF) signal, and other special functions. The other part is an **antenna** for receiving and transmitting the radio signals.

Today's sensor networks are mostly wireless, and are known as wireless sensor networks (WSNs). A typical WSN consists of **spatially** distributed autonomous sensors to **cooperatively** monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion, or **pollutants**. The development of wireless sensor networks was motivated by military applications such as **battlefield** surveillance. WSN technology is now used in many industrial and civilian application areas, including process monitoring and control, machine health monitoring, environment and **habitat** monitoring, health care and home automation, and intelligent traffic control.

The GPS was developed in 1973 by the U.S. Air Force. Similar developments have also occurred in the European Union, Russia, and China. Since 1994, a degraded GPS has been made available for civilian applications in providing reliable positioning, **navigation**, and timing services. For anyone with a GPS receiver, the system will provide accurate location and time information for an unlimited number of users in all weather conditions, day and night, anywhere in the world.

Enabling Technologies	Synergistic Technologies
Machine-to-machine interfaces	Geotagging/geocaching
Protocols of electronic communication	Biometrics
Microcontrollers	Machine vision
Wireless communication	Robotics
RFID	Augmented reality
Energy harvesting technologies	Telepresence and adjustable autonomy
Sensors and sensor networks	Life recorders and personal black boxes
Actuators	Tangible user interfaces
Positioning or location technology (GPS)	Clean technologies
Software engineering	Mirror worlds

Table 9B-1: Enabling and Synergistic Technologies for the IoT

Synergistic technologies play supporting roles. For example, biometrics could be widely applied to personalize the interactions among humans, machines, and objects. Artificial intelligence, computer vision, robotics, and telepresence can make our lives more automated in the future.

actuator

/'æktʃueɪtə /

n. 致动器; 传动装置; 执行机构

tangible /'tæŋdʒəbl /

a. 可触摸的; 有形的; 明确的

software agent

软件代理

fusion /'fju:ʒən /

n. 融合; 融合

miniaturization

/'minɪtʃəraɪ'zeɪʃən;

.minɪtʃəre- /

n. 小型化, 微型化

spectrum

/'spektrəm /

n. 频谱; 系列; 范围

teleoperation

/.telɪ'pə'reɪʃən /

n. 远程操作, 遥操作

geolocation

/.'dʒi:əuleɪ'keɪʃən /

n. 地理位置, 地理定位

diffusion

/dɪ'fju:ʒən /

n. 扩散

expedite

/'ekspɪdaɪt /

v. 迅速执行; 加速

inventory

/'ɪnvəntəri; -tɔ:ri /

v. 为...开列存货清单; 编制...的目录; 盘存

supply chain

供应链

societal /sə'saɪətəl /

a. 社会的

navigator

/'nævɪgeɪtə /

n. 导航仪, 导航系统; 航海者

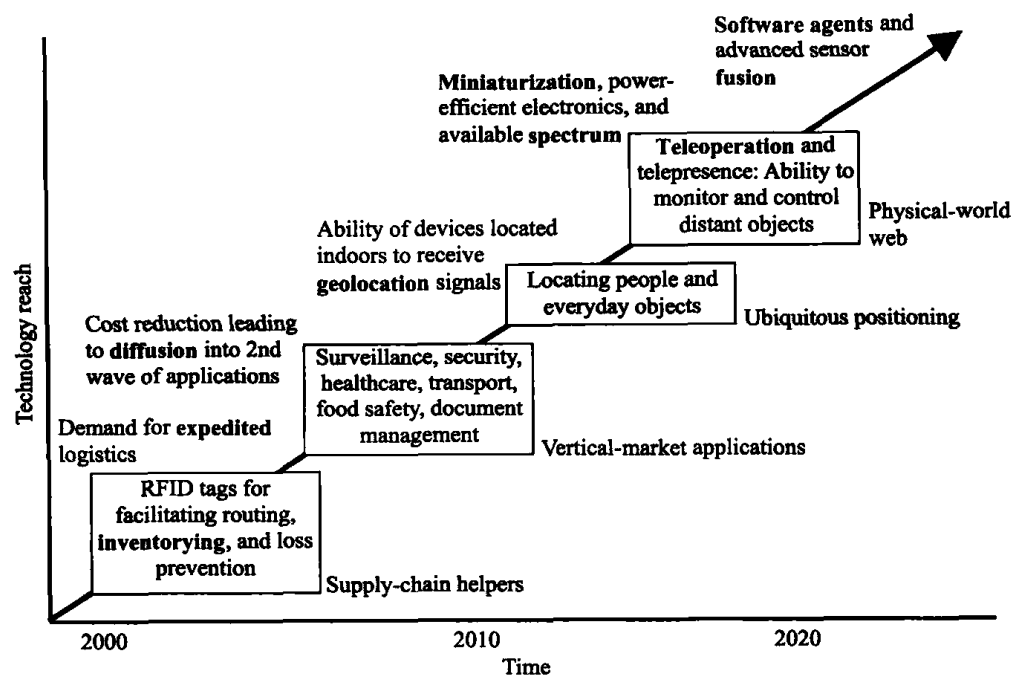


Figure 9B-1: Technology Road Map of the IoT

The IoT development could become more mature and more sophisticated in the next 15 years. Figure 9B-1 shows the major technology advances and key applications that may benefit from the IoT. For example, **supply chains** are now better supported than before. Vertical market applications may represent the next wave of advances. Ubiquitous positioning is expected to become a reality as we move toward 2020. Beyond that, a physical IoT may be in place in a global scale. These advances will significantly upgrade human abilities, **societal** outcomes, national productivity, and quality of life.

III. Architecture of the Internet of Things

The IoT system is likely to have an event-driven architecture. In Figure 9B-2, IoT development is shown with a three-layer architecture. The top layer is formed by driven applications.¹ The application space of the IoT is huge. The bottom layers represent various types of sensing devices: namely RFID tags, ZigBee or other types of sensors, and road-mapping GPS navigators². The sensing devices are locally or wide-area-connected in the form of RFID networks, sensor networks, and GPSes. Signals or information collected at these sensing devices are linked to the applications through the cloud computing platforms at the middle layer.

¹ The top layer is formed by driven applications.: 顶层由受驱动的应用程序构成。

² road-mapping GPS navigators: 标示路线图的 GPS 导航仪。

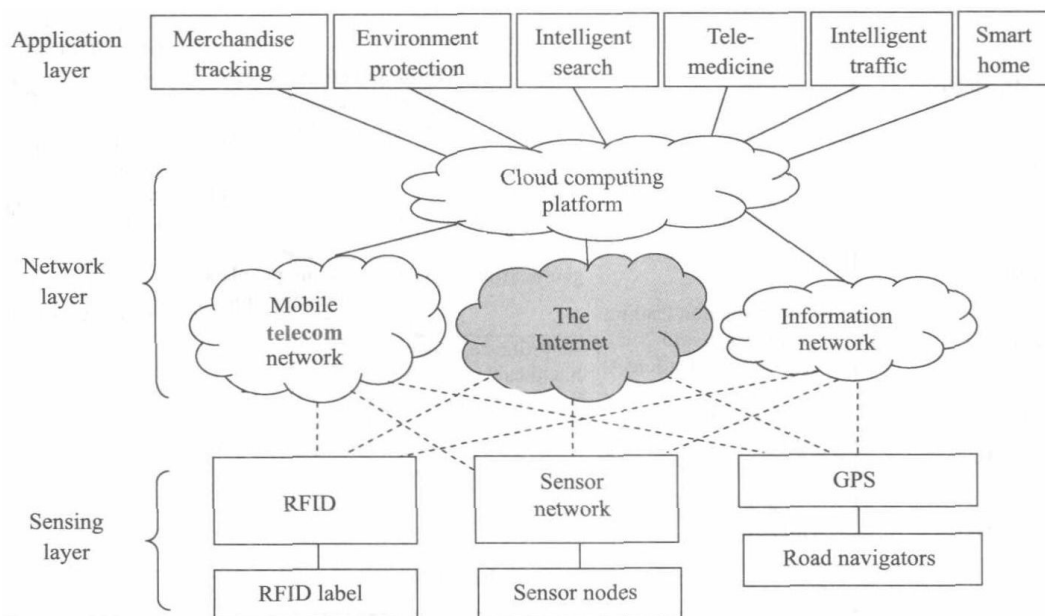


Figure 9B-2: The architecture of an IoT consisting of sensing devices that are connected to various applications via mobile networks, the Internet, and processing clouds

telecom /'telikɒm/
n. 电信 (= telecommunication)
deterministic
/di:tə'mi'nistik/
a. 决定论的; 确定性的

semantic
/si'mæntik/
a. 语义(学)的
Semantic Web
语义网

Semantic Grid
语义网格

avatar /,æve'tɑː/
n. 化身

avoidance
/ə'voidəns/
n. 避免; 避开; 回避

The signal processing clouds are built over the mobile networks, the Internet backbone, and various information networks at the middle layer. In the IoT, the meaning of a sensing event does not follow a **deterministic** or syntactic model.¹ In fact, the service-oriented architecture (SOA) model is adoptable here. A large number of sensors and filters are used to collect the raw data. Various compute and storage clouds and grids are used to process the data and transform it into information and knowledge formats. The sensed information is used to put together a decision-making system for intelligence applications. The middle layer is also considered as a **Semantic Web** or **Grid**. Some actors (services, components, **avatars**) are self-referenced.²

IV. Applications of the Internet of Things

Table 9B-2 summarizes IoT applications in three major civilian application domains. Obviously, the IoT has a lot of military applications, which is beyond the scope of this section. In general, use of the IoT aims to promote industrial productivity and raise economic growth. The IoT plays important roles in environment protection, including pollution control, weather forecasting, and disaster **avoidance** and recovery. In terms of

¹ *In the IoT, the meaning of a sensing event does not follow a deterministic or syntactic model.*: 在物联网中, 一个感知事件的意义不是遵循确定性或句法模型的。

² *Some actors (services, components, avatars) are self-referenced.*: 有些行为体(服务、组件、化身)是自引用的。

beneficiary

/ˌbenɪˈfɪjəri/

n. 受益人, 受惠者

indicative

/ɪnˈdɪkətɪv/

a. 标示的; 指示的

logistics

/ləˈdʒɪstɪks/

n. 后勤(学), 物流

retailing /ˈriːteɪlɪŋ/

n. 零售业

retailer /ˈriːteɪlə/

n. 零售商; 零售店

unattended

/ˌʌnəˈtendɪd/

a. 没人看管的; 没人照料的

checkout /ˈtʃekaut/

n. 结账(离去);

付款台

shoplifting

/ˈʃɒplɪftɪŋ/

n. (混在顾客群中进行的) 商店货物扒窃

cold chain 冷链**audit** /ˈɔːdɪt/

v. 审计, 查账

assurance /əˈʃʊərəns/

n. 保证; 把握, 信心

perishable

/ˈperɪʃəbəl/

a. 易腐的, 易烂的

intact /ɪnˈtækt/

a. 完整无缺的; 未受损伤的

entail /ɪnˈteɪl/

v. 使成为必要, 需要

refrigeration

/reˌfrɪdʒəˈreɪʃən/

n. 冷藏; 制冷

data logging

数据(事件)记录;

数据登录

participant

/pɑːˈtɪsɪpənt/

n. 参加者, 参与者

societal impacts, the IoT can make our lives more convenient and comfortable. Government services, law enforcement, and home and health improvements are the major **beneficiaries**. In the remaining space of this section, we will briefly discuss some of the application domains.

Domain	Brief Descriptions	Indicative Examples
Growth in industry and economy	Activities involving financial or commercial transactions between companies and organizations	Manufacturing, logistics , service sector, banking, financial governmental authorities, intermediaries, etc.
Environment and natural resources	Activities regarding the protection, monitoring, and development of all natural resources	Agriculture and breeding, recycling, environmental management services, energy management, etc.
Society and daily life	Activities/initiatives regarding the development and inclusion of societies, cities, and people	Governmental services toward citizens and society structures, e-inclusion (aging, disabled people), etc. ¹

Table 9B-2: Some Application Domains of the IoT

1. Retailing and Logistics Services

Emergence of RFID applications depends strongly on adoption by **retailers**, logistics organizations, and package-delivery companies. In particular, retailers may tag individual objects in order to solve a number of problems at once: accurate inventorying, loss control, and ability to support **unattended** walk-through point of sale terminals (which promise to speed **checkout** while reducing both **shoplifting** and labor costs). **Cold-chain auditing** and **assurance**² could require tagging food and medicine with temperature-sensitive materials and/or electronics; ensuring or monitoring whether **perishable** materials are **intact** and/or need attention may **entail** communications among things, **refrigeration** systems, automated **data logging** systems, and human technicians.

2. Supply Chain Management

Supply chain management is a process used by companies to ensure that their supply chain is efficient and cost-effective. Supply chain management can be aided by an IoT system. The idea is to manage a whole network of related businesses or partners involved in product manufacturing, delivery, and services as required by end customers. At any given time, market forces could demand changes from suppliers, logistics providers, locations and customers, and any number of specialized **participants** in a

¹ Governmental services toward citizens and society structures, e-inclusion (aging, disabled people), etc.: 政府对公民的服务与社会结构、电子包容(衰老、残疾人士)等。

² Cold-chain auditing and assurance: 冷链审计与保证。

workflow /'we:kfleu/

n. workflow

grid /grid/

n. 系统网络(指输电线路、广播电台、电视台、天然气管道等网络)

power grid 电网**thermostat**

/'θə:meustæt/

n. 温度自动调节器, 恒温器

residential

/rezi'denʃəl/

a. 居住的; 住宅的

mall /mɔ:l, mæl/

n. (非露天的) 购物区, 购物街

shopping mall

(同一建筑物里有许多商店的) 购物中心

high-rise /'haɪ'reɪz/

a. (建筑) 高层的

office tower

办公大楼

suppression

/sə'preʃən/

n. 压制; 抑制

supply chain. This variability has significant effects on the supply chain infrastructure, ranging from the foundation layers of establishing the electronic communication between the trading partners to the more complex configuration of the processes and the arrangement of **workflows** that are essential to a fast production process.

3. Smart Power Grid and Smart Buildings

A critical IoT application is promotion of a smart power grid. Various power companies across the United States have or are in the process of upgrading their power management and distribution systems. Various sensors at individual homes (smart **thermostats**) can collect information that is sent via a network to main stations (perhaps even local “hubs”) that can apply complex power management and send control signals back to the grid to save energy. The smart grid is made possible by applying sensing, measurement, and control devices to electricity production, transmission, distribution, and consumption.

The IoT has been suggested in construction of smart buildings in **residential**, commercial, industrial, and government settings. A smart building can be a **shopping mall** or a home, a hospital or a **high-rise office tower**. Smart buildings need monitoring and regulation of heating, air conditioning, lighting, and environmental changes. They can oversee building security, fire **suppression**, and elevator operations. Smart building technologies focus on bringing more detailed monitoring and sensing “awareness” to buildings.

Exercises

I. Fill in the blanks with the information given in the text:

- Supportive technologies for the IoT are divided into two categories: _____ and synergistic technologies.
- Contemporary devices that lend support to ubiquitous computing include smartphones, _____ computers, sensor networks, RFID tags, _____ cards, and GPS devices.
- The architecture of an IoT consists of _____ devices connected to various applications via mobile networks, the Internet, and processing

_____.

4. A typical wireless sensor network consists of spatially distributed _____ sensors to cooperatively monitor physical or environmental conditions.

II. Translate the following terms or phrases from English into Chinese and vice versa:

- | | |
|-------------------------|-----------|
| 1. computer vision | 6. 移动网络 |
| 2. ubiquitous computing | 7. 平板电脑 |
| 3. command line | 8. 物联网 |
| 4. data logging system | 9. 智能电网 |
| 5. augmented reality | 10. 传感器融合 |

Section C

Mobile Users



tweet /twi:t/
v. 上推特, 发微博
surf /sɜ:f/
v. (在...) 冲浪
blanket /'blæŋkit/
v. 用毯子(或毯状物)盖(或裹); 覆在...的上面
mobile phone
移动电话
patchwork
/'pætʃwɜ:k/
n. 拼缀物; 拼凑的东西, 杂烩

Mobile computers, such as laptop and handheld computers, are one of the fastest-growing segments of the computer industry. Their sales have already overtaken those of desktop computers. Why would anyone want one? People on the go¹ often want to use their mobile devices to read and send email, **tweet**, watch movies, download music, play games, or simply to **surf** the Web for information. They want to do all of the things they do at home and in the office. Naturally, they want to do them from anywhere on land, sea or in the air.

Connectivity to the Internet enables many of these mobile uses. Since having a wired connection is impossible in cars, boats, and airplanes, there is a lot of interest in wireless networks. Cellular networks operated by the telephone companies are one familiar kind of wireless network that **blankets** us with coverage for **mobile phones**. Wireless hotspots based on the 802.11 standard are another kind of wireless network for mobile computers. They have sprung up everywhere that people go, resulting in a **patchwork** of coverage at cafes, hotels, airports, schools, trains and planes. Anyone with a laptop computer and a wireless modem can just turn on their

¹ **on the go**: 〈口〉忙个不停。这里指“在旅行中”。

computer and be connected to the Internet through the hotspot, as though the computer were plugged into a wired network.

Wireless networks are of great value to fleets of trucks, taxis, delivery vehicles, and repairpersons for keeping in contact with their home base. For example, in many cities, taxi drivers are independent businessmen, rather than being employees of a taxi company. In some of these cities, the taxis have a display the driver can see. When a customer calls up, a central dispatcher types in the pickup and destination points. This information is displayed on the drivers' displays and a beep sounds. The first driver to hit a button on the display gets the call.

Wireless networks are also important to the military. If you have to be able to fight a war anywhere on Earth at short notice, counting on using the local networking infrastructure is probably not a good idea. It is better to bring your own.

Although wireless networking and mobile computing are often related, they are not identical, as Figure 9C-1 shows. Here we see a distinction between fixed wireless and mobile wireless networks. Even notebook computers are sometimes wired. For example, if a traveler plugs a notebook computer into the wired network jack in a hotel room, he has mobility without a wireless network.

Wireless	Mobile	Typical Applications
No	No	Desktop computers in offices
No	Yes	A notebook computer used in a hotel room
Yes	No	Networks in unwired buildings
Yes	Yes	Store inventory with a handheld computer

Figure 9C-1: Combinations of Wireless Networks and Mobile Computing

Conversely, some wireless computers are not mobile. In the home, and in offices or hotels that lack suitable cabling, it can be more convenient to connect desktop computers or media players wirelessly than to install wires. Installing a wireless network may require little more than¹ buying a small box with some electronics in it, unpacking it, and plugging it in. This solution may be far cheaper than having workmen put in cable ducts to wire the building.

Finally, there are also true mobile, wireless applications, such as

dispatcher

/dis'pætʃə/

n. (车辆) 调度员

pickup /'pikʌp/

n. 搭车; 接人; 提货

beep /bi:p/

n. 短促而尖利的声音, 嘟

notebook

/'neutbuk/

n. 笔记本 (式计算机)

notebook computer

笔记本

jack /dʒæk/

n. 插座; 插口

inventory

/'inventəri;-tɔ:ri/

n. 存货 (清单), 库存

cabling /'keiblin/

n. 电缆

duct /dʌkt/

n. 管道; 导管

¹ *little more than*: 只是...而已; 只有, 仅仅。

rental /'rentəl/

n. 租赁, 出租

parking lot

〈美〉停车场, 停车区

barcode /'bɑ:kəud/

n. 条形码

texting /'tekstɪŋ/

n. 发短信

subscriber

/səb'skraɪbə/

n. 用户, 订户

Short Message Service

短信(服务)

profitable

/'prɒfɪtəbl/

a. 有利可图的, 有盈利的

carrier /'kæriə/

n. 运输公司; 运输工具; 通信公司

consumer**electronics**

消费电子产品

electronic book**reader**

电子(图)书阅读器

roam /ru:m/

v. 漫步; 漫游

electronic picture**frame**

电子相框

people walking around stores with handheld computers recording inventory. At many busy airports, car rental return clerks work in the parking lot with wireless mobile computers. They scan the barcodes or RFID chips of returning cars, and their mobile device, which has a built-in printer, calls the main computer, gets the rental information, and prints out the bill on the spot.

Perhaps the key driver of mobile, wireless applications is the mobile phone. Text messaging or texting is tremendously popular. It lets a mobile phone user type a short message that is then delivered by the cellular network to another mobile subscriber. Few people would have predicted ten years ago that having teenagers tediously typing short text messages on mobile phones would be an immense money maker for telephone companies. But texting (or Short Message Service as it is known outside the U.S.) is very profitable since it costs the carrier but a tiny fraction of one cent to relay a text message, a service for which they charge far more.

The long-awaited convergence of telephones and the Internet has finally arrived, and it will accelerate the growth of mobile applications. Smart phones, such as the popular iPhone, combine aspects of mobile phones and mobile computers. The (3G¹ and 4G²) cellular networks to which they connect can provide fast data services for using the Internet as well as handling phone calls. Many advanced phones connect to wireless hotspots too, and automatically switch between networks to choose the best option for the user.

Other consumer electronics devices can also use cellular and hotspot networks to stay connected to remote computers. Electronic book readers can download a newly purchased book or the next edition of a magazine or today's newspaper wherever they roam. Electronic picture frames can update their displays on cue³ with fresh images.

Since mobile phones know their locations, often because they are equipped with GPS (Global Positioning System) receivers, some services are intentionally location dependent. Mobile maps and directions are an obvious candidate⁴ as your GPS-enabled phone and car probably have a

¹ 3G: 第三代移动通信技术 (3rd Generation 的缩略)。

² 4G: 第四代移动通信技术 (4th Generation 的缩略)。

³ on cue: 恰好在这时候; 恰好在得到提示的那一刻; 接受了提示(或暗示)。在句中意思相当于“根据信号(或指令)”。

⁴ Mobile maps and directions are an obvious candidate: 移动地图与路线是显而易见的适选对象。

annotate

/ˈæneuteɪt/

v. (给...)作注解(或评注)

annotation

/ˌæneuˈteɪʃən/

n. 注解, 注释, 评注

authorize /ˈɔ:θəraɪz/

v. 授权; 委托

vend /vend/

v. 出售(尤指沿街叫卖零售); 用售货机出售

vending machine

投币式自动售货机

credit card

信用卡

smartcard

/ˈsmɑ:tka:d/

n. 智能卡

backfire /ˌbækˈfaɪə/

v. 发生意外, 产生事与愿违的结果

howl /haul/

v. 吼叫; 怒吼

cash register

现金出纳机, 收银机

tack /tæk/v. 附加, 增补, 追加
(on)

better idea of where you are than you do. So, too, are searches for a nearby bookstore or Chinese restaurant, or a local weather forecast. Other services may record location, such as **annotating** photos and videos with the place at which they were made. This **annotation** is known as “geo-tagging.”

An area in which mobile phones are now starting to be used is m-commerce (mobile-commerce). Short text messages from the mobile are used to **authorize** payments for food in **vending machines**, movie tickets, and other small items instead of cash and **credit cards**. The charge then appears on the mobile phone bill. When equipped with NFC¹ (Near Field Communication) technology the mobile can act as an RFID **smartcard** and interact with a nearby reader for payment. The driving forces behind this phenomenon are the mobile device makers and network operators, who are trying hard to figure out how to get a piece of the e-commerce pie. From the store's point of view, this scheme may save them most of the credit card company's fee, which can be several percent. Of course, this plan may **backfire**, since customers in a store might use the RFID or barcode readers on their mobile devices to check out competitors' prices before buying and use them to get a detailed report on where else an item can be purchased nearby and at what price.

One huge thing that m-commerce has going for it² is that mobile phone users are accustomed to paying for everything (in contrast to Internet users, who expect everything to be free). If an Internet Web site charged a fee to allow its customers to pay by credit card, there would be an immense **howling** noise from the users. If, however, a mobile phone operator allowed its customers to pay for items in a store by waving the phone at the **cash register** and then **tacked** on a fee for this convenience, it would probably be accepted as normal. Time will tell.³

No doubt the uses of mobile and wireless computers will grow rapidly in the future as the size of computers shrinks, probably in ways no one can now foresee. Let us take a quick look at some possibilities. Sensor networks are made up of nodes that gather and wirelessly relay information they sense about the state of the physical world. The nodes may be part of familiar items such as cars or phones, or they may be small separate devices.

¹ NFC: 近场通信, 近距离无线通信 (Near Field Communication 的首字母缩略)。

² *One huge thing that m-commerce has going for it*: 移动电子商务拥有的一个巨大优势。that m-commerce has going for it 是一个定语从句, 修饰 one huge thing, one huge thing 作 has 的宾语, it 指 m-commerce (相当于 m-commerce has one huge thing going for it)。

³ *Time will tell*: 时间会证明一切的。

on-board

/'ɒn'bɔ:d, 'ɔ:n-/

a. 车载的; 机载的;
舰载的**diagnostic**

/,daɪəɡ'nɒstɪk/

a. 诊断的

upload /ˌʌp'ləʊd/

v. 上传, 上载

pothole /'pɒθəʊl/

n. 路面凹坑, 坑洞

congest /kən'dʒest/

v. 拥挤; 拥塞

guzzler /'gʌzlə/n. 狂饮者; 滥吃者;
大量消耗者**gas guzzler**耗油量大的汽车,
油老虎**revolutionize**

/ˌrevə'ju:ʃənaɪz/

v. 使发生革命性剧变

zebra /'zi:brə, 'ze-/

n. 斑马

rodent /'rəʊdənt/n. 啮齿目动物 (如
鼠、松鼠、河狸等)**mundane**

/mʌn'deɪn/

a. 世俗的; 平凡的

parking meter汽车停放计时器,
汽车停放收费计**debit** /'deɪt/

n. 借方; 借记, 借入

debit card

借记卡, 借方卡

For example, your car might gather data on its location, speed, vibration, and fuel efficiency from its on-board diagnostic system and upload this information to a database. Those data can help find potholes, plan trips around congested roads, and tell you if you are a “gas guzzler” compared to other drivers on the same stretch of road.

Sensor networks are revolutionizing science by providing a wealth of data on behavior that could not previously be observed. One example is tracking the migration of individual zebras by placing a small sensor on each animal. Researchers have packed a wireless computer into a cube 1 mm on edge.¹ With mobile computers this small,² even small birds, rodents, and insects can be tracked.

Even mundane uses, such as in parking meters, can be significant because they make use of data that were not previously available. Wireless parking meters can accept credit or debit card payments with instant verification over the wireless link. They can also report when they are in use over the wireless network. This would let drivers download a recent parking map to their car so they can find an available spot more easily. Of course, when a meter expires, it might also check for the presence of a car (by bouncing a signal off it) and report the expiration to parking enforcement.

Wearable computers are another promising application. Smart watches with radios have been part of our mental space since their appearance in the Dick Tracy³ comic strip in 1946;⁴ now you can buy them. Other such devices may be implanted, such as pacemakers and insulin pumps. Some of these can be controlled over a wireless network. This lets doctors test and reconfigure them more easily. It could also lead to some nasty problems if the devices are as insecure as the average PC and can be hacked easily.

¹ *Researchers have packed a wireless computer into a cube 1 mm on edge.*: 研究人员已经将一个无线计算机装入一个边长为 1 毫米的立方体中。

² *With mobile computers this small*: 有了这么小的移动计算机。this 可作副词, 与形容词或副词连用, 表示“达到这样的程度”、“这样”、“这么”。

³ *Dick Tracy*: 《迪克·特雷西》, 一部美国长篇连载漫画, 描绘虚构人物迪克·特雷西, 一个智勇双全、枪法奇快神准的天才警探, 与形形色色的歹徒斗智斗勇的传奇故事。由漫画家切斯特·古尔德 (Chester Gould) 创作, 1931 年 10 月 4 日在《底特律镜报》(Detroit Mirror) 上首次刊载。古尔德一直执笔创作《迪克·特雷西》到 1977 年。该漫画 1990 年被翻拍为电影, 亦译为《至尊神探》。

⁴ *Smart watches with radios have been part of our mental space since their appearance in the Dick Tracy comic strip in 1946*: 带有收音机的智能手表自 1946 年出现在《迪克·特雷西》连环漫画中以来, 一直是我们梦寐以求之物。

expire / ik'spaɪə /

v. 满期, 届期

expiration

/ ,ɛkspɪ'reɪʃən /

n. 满期, 届期

wearable computer

可穿戴计算机,

穿戴式计算机

comic strip (报刊上的) 连环漫画

implant

/ im'plɑ:nt; -'plænt /

v. 植入, 埋置

pacemaker

/ 'peɪs,meɪkə /

n. 起搏器

insulin

/ 'ɪnsjʊlɪn; 'ɪnsə- /

n. 胰岛素 (制剂)

insulin pump

胰岛素泵

hack / hæk /

v. 非法闯入 (计算机网络), 黑客攻击

Exercises

I. Fill in the blanks with the information given in the text:

1. The sales of mobile computers have already overtaken those of _____ computers.
2. Perhaps the key driving force behind mobile, _____ applications is the mobile phone.
3. The _____ of sensor networks may be part of familiar items such as cars or phones, or they may be small separate devices.
4. Combining aspects of mobile _____ and mobile _____, smart phones can use the Internet as well as make phone calls.

II. Translate the following terms or phrases from English into Chinese and vice versa:

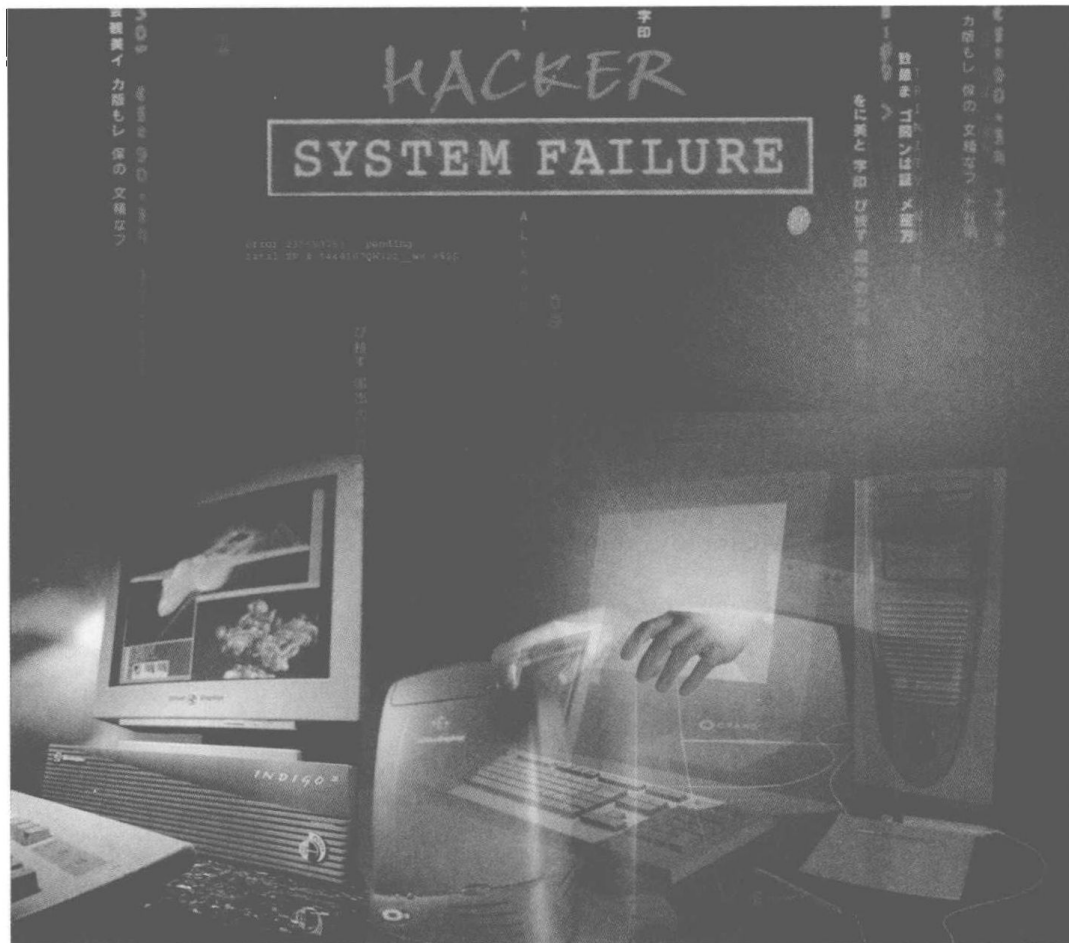
- | | |
|--------------------------|-----------|
| 1. notebook computer | 5. 移动电话 |
| 2. wireless hotspot | 6. 条形码阅读器 |
| 3. Short Message Service | 7. 网站 |
| 4. wearable computer | 8. 智能手机 |

Unit 10 Computer Security

(计算机安全)

Section A

Computer Security



I. A Definition of Computer Security

The NIST¹ Computer Security Handbook defines *computer security* as follows:

confidentiality
/ 'kɒnfi.dənʃi'æliti /
n. 机密性

The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability, and **confidentiality** of information system resources (including hardware, software, firmware,

¹ NIST: (美国) 国家标准与技术研究院 (National Institute of Standards and Technology 的首字母缩略)。

information/data, and telecommunications).

This definition introduces three key objectives that are at the heart of computer security:

- **Confidentiality:** This term covers two related concepts:
 - **Data confidentiality:** Assures that private or confidential information is not made available or disclosed to unauthorized individuals.
 - **Privacy:** Assures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed.¹
- **Integrity:** This term covers two related concepts:
 - **Data integrity:** Assures that information and programs are changed only in a specified and authorized manner.
 - **System integrity:** Assures that a system performs its intended function in an **unimpaired** manner, free from deliberate or inadvertent unauthorized manipulation of the system.
- **Availability:** Assures that systems work promptly and service is not denied to authorized users.

These three concepts form what is often referred to as the CIA² **triad**. Although the use of the CIA triad to define security objectives is well established, some in the security field feel that additional concepts are needed to present a complete picture. Two of the most commonly mentioned are as follows:

- **Authenticity:** The property of being genuine and being able to be verified and trusted; confidence in the **validity** of a transmission, a message, or a message **originator**. This means verifying that users are who they say they are and that each input arriving at the system came from a trusted source.
- **Accountability:** The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity. This supports **nonrepudiation**, **deterrence**, **fault isolation**, intrusion detection and prevention, and after-action recovery and legal action.

unimpaired

/ˌʌnimˈpeəd/

a. 未受损的;
未削弱的

triad /ˈtraɪəd, -æd/

n. 三件一套 (或
组); 三位一体

authenticity

/ˌɔːθənˈtɪsɪti/

n. 可靠性, 真实性

validity /veˈlɪdɪti/

n. 有效 (性)

originator

/əˈrɪdʒɪneɪtə/

n. 创始人; 起源

accountability

/əˌkaʊntəˈbɪlɪti/

n. 负有责任

nonrepudiation

/ˈnɒnriˌpjʊːdiˈeɪʃən/

n. 不拒绝; 不否认;
不可抵赖

deterrence

/diˈterəns/

n. 威慑

fault isolation

故障隔离

¹ Assures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed.: 确保个人对哪些与其有关的信息可被收集和存储, 以及该信息可由谁透露和透露给谁予以控制或影响。

² CIA: 机密性、完整性与可用性 (confidentiality, integrity, and availability 的首字母缩略)。

breach /bri:tʃ/
n. 破坏; 违反
forensic /fə'rensɪk/
a. (用于) 法庭的;
法医的

Because truly secure systems aren't yet an achievable goal, we must be able to trace a security **breach** to a responsible party. Systems must keep records of their activities to permit later **forensic** analysis to trace security breaches or to aid in transaction disputes.

II. Threats and Assets

The assets of a computer system can be categorized as hardware, software, data, and communication lines and networks. We will briefly describe these four categories and relate them to the concepts of integrity, confidentiality, and availability.

1. Hardware

susceptible
/sə'septəbl/
a. 易受影响的 (*to*)
proliferation
/prəʊˌlɪfə'reɪʃən/
n. 激增; 扩散

A major threat to hardware is the threat to availability. Hardware is the most vulnerable to attack and the least **susceptible** to automated controls. Threats include accidental and deliberate damage to equipment as well as theft. The **proliferation** of personal computers and workstations and the widespread use of LANs increase the potential for losses in this area. Theft of CD-ROMs and DVDs can lead to loss of confidentiality. Physical and administrative security measures are needed to deal with these threats.

2. Software

piracy /'paɪərəsi/
n. 侵犯版权, 盗版, 剽窃
countermeasure
/'kauntəˌmeʒə/
n. 对策, 对抗手段

A key threat to software is an attack on availability. Software, especially application software, is often easy to delete. Software can also be altered or damaged to render it useless. Careful software configuration management, which includes making backups of the most recent version of software, can maintain high availability. A more difficult problem to deal with is software modification that results in a program that still functions but behaves differently than before, which is a threat to integrity/authenticity. Computer viruses and related attacks fall into this category. A final problem is protection against software **piracy**. Although certain **countermeasures** are available, by and large¹ the problem of unauthorized copying of software has not been solved.

3. Data

secrecy /'si:kresi/
n. 秘密; 保密

Security concerns with respect to data are broad, encompassing availability, **secrecy**, and integrity. In the case of availability, the concern is

¹ *by and large*: 大体上, 总的说来, 一般地说。

with the destruction of data files, which can occur either accidentally or maliciously.

The obvious concern with secrecy is the unauthorized reading of data files or databases, and this area has been the subject of perhaps more research and effort than any other area of computer security. A less obvious threat to secrecy involves the analysis of data and manifests itself in the use of so-called statistical databases, which provide summary or aggregate information. As the use of statistical databases grows, there is an increasing potential for **disclosure** of personal information. For example, if one table records the **aggregate** of the incomes of **respondents** A, B, C, and D and another records the aggregate of the incomes of A, B, C, D, and E, the difference between the two aggregates would be the income of E.

Finally, data integrity is a major concern in most installations. Modifications to data files can have consequences ranging from minor to disastrous.

4. Communication Lines and Networks

Network security attacks can be classified as *passive attacks* and *active attacks*. A passive attack attempts to learn or make use of information from the system but does not affect system resources. An active attack attempts to alter system resources or affect their operation.

Passive attacks are in the nature of **eavesdropping** on, or monitoring of, transmissions. The goal of the attacker is to obtain information that is being transmitted. Two types of passive attacks are release of message contents and traffic analysis.

The **release of message contents** is easily understood. A telephone conversation, an electronic mail message, and a transferred file may contain sensitive or confidential information. We would like to prevent an opponent from learning the contents of these transmissions.

Traffic analysis is subtler. If we had encryption protection in place, an opponent might still be able to observe the pattern of messages. The opponent could determine the location and identity of communicating hosts and could observe the frequency and length of messages being exchanged. This information might be useful in guessing the nature of the communication that was taking place.

Passive attacks are very difficult to detect because they do not involve

disclosure

/dis'kləʊʒə/

n. 泄露, 透露

aggregate

/'ægrɪget/

n. 总数, 合计; 聚集体

respondent

/rɪ'spɒndənt/

n. 〈主美〉(民意测验等的)调查对象, (调查表、问题表的)答卷人

eavesdrop

/'i:vzdrɒp/

v. 偷听, 窃听 (*on*)

any alteration of the data. Thus, the emphasis in dealing with passive attacks is on prevention rather than detection.

Active attacks involve some modification of the data stream or the creation of a false stream and can be subdivided into four categories: replay, **masquerade**, modification of messages, and denial of service.

Replay involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect.

A **masquerade** takes place when one entity pretends to be a different entity. A masquerade attack usually includes one of the other forms of active attack. For example, authentication sequences can be captured and replayed after a valid authentication sequence has taken place, thus enabling an authorized entity with few privileges to obtain extra privileges by **impersonating** an entity that has those privileges.

Modification of messages simply means that some portion of a **legitimate** message is altered, or that messages are delayed or reordered, to produce an unauthorized effect.

The **denial of service** prevents or **inhibits** the normal use or management of communications facilities. This attack may have a specific target; for example, an entity may **suppress** all messages directed to a particular destination. Another form of service denial is the **disruption** of an entire network, either by disabling the network or by **overloading** it with messages so as to **degrade** performance.

It is quite difficult to prevent active attacks absolutely, because to do so would require physical protection of all communications facilities and paths at all times. Instead, the goal is to detect them and to recover from any disruption or delays caused by them.

III. Computer Security Strategy

A comprehensive security strategy involves three aspects:

- **Specification/policy:** What is the security scheme supposed to do?
- **Implementation/mechanisms:** How does it do it?
- **Correctness/assurance:** Does it really work?

1. Security Policy

In developing a security policy, a security manager needs to consider the following factors:

masquerade

/ˌmæskə'reɪd, ˌmɑːs-/
n. 假面舞会; 伪装;
假扮

impersonate

/ɪm'pɜːsəneɪt/
v. 扮演; 模仿;
假冒

legitimate

/lɪ'dʒɪtɪmət/
a. 合法的; 正当的

inhibit

/ɪn'hɪbɪt/
v. 抑制, 约束

suppress

/sə'pres/
v. 压制; 抑制;
阻止

disruption

/dɪs'rʌpʃən/
n. 扰乱; 中断

overload

/ˌəʊvə'ləʊd/
v. 使过载, 使超载

degrade

/dɪ'greɪd/
v. 使降级, 使退化

vulnerability

/ˌvʌlnərə'bɪlɪti/

n. 易遭攻击的地方; 脆弱(性)

likelihood

/'laɪklɪhʊd/

n. 可能, 可能性

versus /'vɜ:səs/prep. 对; 与...相对
(或相比)**system crash**

系统崩溃

monetary

/'mʌnɪtəri/

a. 货币的; 金钱的

encryption key

(加密) 密钥

log /'lɒg/v. 登录, 注册,
进入系统 (*on, in*)

- The value of the assets being protected
- The vulnerabilities of the system
- Potential threats and the likelihood of attacks

Further, the manager must consider the following tradeoffs:

- **Ease of use versus security:** Virtually all security measures involve some penalty in the area of ease of use. For example, virus-checking software reduces available processing power and introduces the possibility of system crashes or malfunctions due to improper interaction between the security software and the operating system.
- **Cost of security versus cost of failure and recovery:** In addition to ease of use and performance costs, there are direct monetary costs in implementing and maintaining security measures. All of these costs must be balanced against the cost of security failure and recovery if certain security measures are lacking.

2. Security Implementation

Security implementation involves four complementary courses of action:

- **Prevention:** There is a wide range of threats in which prevention is a reasonable goal. For example, if a secure encryption algorithm is used, and if measures are in place to prevent unauthorized access to encryption keys, then attacks on confidentiality of the transmitted data will be prevented.
- **Detection:** In a number of cases, absolute protection is not feasible, but it is practical to detect security attacks. For example, there are intrusion detection systems designed to detect the presence of unauthorized individuals logged onto a system.
- **Response:** If security mechanisms detect an ongoing attack, the system may be able to respond in such a way as to halt the attack and prevent further damage.
- **Recovery:** An example of recovery is the use of backup systems, so that if data integrity is compromised, a prior, correct copy of the data can be reloaded.

3. Assurance and Evaluation

Those who are “consumers” of computer security services and mechanisms desire a belief that the security measures in place work as intended. That is, security consumers want to feel that the security infrastructure of their systems meet security requirements and enforce security policies. These considerations bring us to the concepts of assurance and evaluation.

Assurance is the degree of confidence one has that the security measures, both technical and operational, work as intended to protect the system and the information it processes. This encompasses both system design and system implementation. Thus, assurance deals with the questions, “Does the security system design meet its requirements?” and “Does the security system implementation meet its specifications?” Note that assurance is expressed as a degree of confidence, not in terms of a formal proof that a design or implementation is correct. With the present **state of the art**, it is very difficult if not impossible to move beyond a degree of confidence to absolute proof.

state of the art

(学科、技术等当前的或某一时期的) 发展水平, 最新水平

thrust / θrʌst /

n. 要点, 要旨; 目标

Evaluation is the process of examining a computer product or system with respect to certain criteria. Evaluation involves testing and may also involve formal analytic or mathematical techniques. The central **thrust** of work in this area is the development of evaluation criteria that can be applied to any security system (encompassing security services and mechanisms) and that are broadly supported for making product comparisons.¹

Exercises

I. Fill in the blanks with the information given in the text:

1. The CIA triad used to define computer security objectives consists of _____, integrity, and _____.
2. In computer security, integrity covers the two related concepts of _____ integrity and system integrity.

¹ *that are broadly supported for making product comparisons*: 在进行产品比较方面得到广泛支持。evaluation criteria 后面跟有两个定语从句, 这是其中之一。关系代词 that 在两个定语从句中均为主语, 与谓语动词属于被动关系。

3. In addition to the CIA triad, two of the most commonly mentioned concepts concerning security objectives are _____ and accountability.

4. Security concerns with respect to data encompass availability, secrecy, and _____.

5. _____ network security attacks attempt to learn or make use of information from the system without affecting system resources, whereas _____ attacks attempt to alter system resources or affect their operation.

6. A key threat to software is an attack on _____.

7. In developing a security policy, a security manager must balance _____ of use against security, and cost of _____ against cost of failure and recovery.

8. In the context of computer security, _____ is the degree of confidence one has that the security measures work as intended.

II. Translate the following terms or phrases from English into Chinese and vice versa:

- | | |
|-------------------------------|-------------|
| 1. backup system | 11. 软件版本 |
| 2. encryption key | 12. 数据完整性 |
| 3. data confidentiality | 13. 系统崩溃 |
| 4. system vulnerability | 14. 病毒检查软件 |
| 5. unauthorized access | 15. 综合安全策略 |
| 6. intrusion detection system | 16. 软件配置管理 |
| 7. after-action recovery | 17. 故障隔离 |
| 8. software piracy | 18. 统计数据库 |
| 9. authorized user | 19. 保密的加密算法 |
| 10. data unit | 20. 数据流 |

III. Fill in each of the blanks with one of the words given in the following list, making changes if necessary:

- | | | | |
|-----------------------|------------------|------------------|------------------|
| <i>database</i> | <i>control</i> | <i>allow</i> | <i>utility</i> |
| <i>user</i> | <i>native</i> | <i>resource</i> | <i>determine</i> |
| <i>authentication</i> | <i>system</i> | <i>operating</i> | <i>record</i> |
| <i>access</i> | <i>component</i> | <i>robust</i> | <i>function</i> |

An access control mechanism mediates between a user (or a process executing on behalf of a user) and system _____, such as applications, operating systems, firewalls, routers, files, and databases. The _____ must first *authenticate* (验证) a user seeking access. Typically the _____ function determines whether the user is permitted to _____ the system at all. Then the access control _____ determines if the specific requested access by this _____ is permitted. A security administrator maintains an *authorization* (授权) _____ that specifies what type of access to which resources is _____ for this user. The access control function consults this database to _____ whether to grant access. An auditing function monitors and keeps a _____ of user accesses to system resources.

In practice, a number of _____ may cooperatively share the access control function. All _____ systems have at least a *rudimentary* (基本的), and in many cases a quite _____, access control component. Add-on security packages can add to the _____ access control capabilities of the OS. Particular applications or _____, such as a database management system, also incorporate access _____ functions. External devices, such as firewalls, can also provide access control services.

IV. Translate the following passage from English into Chinese:

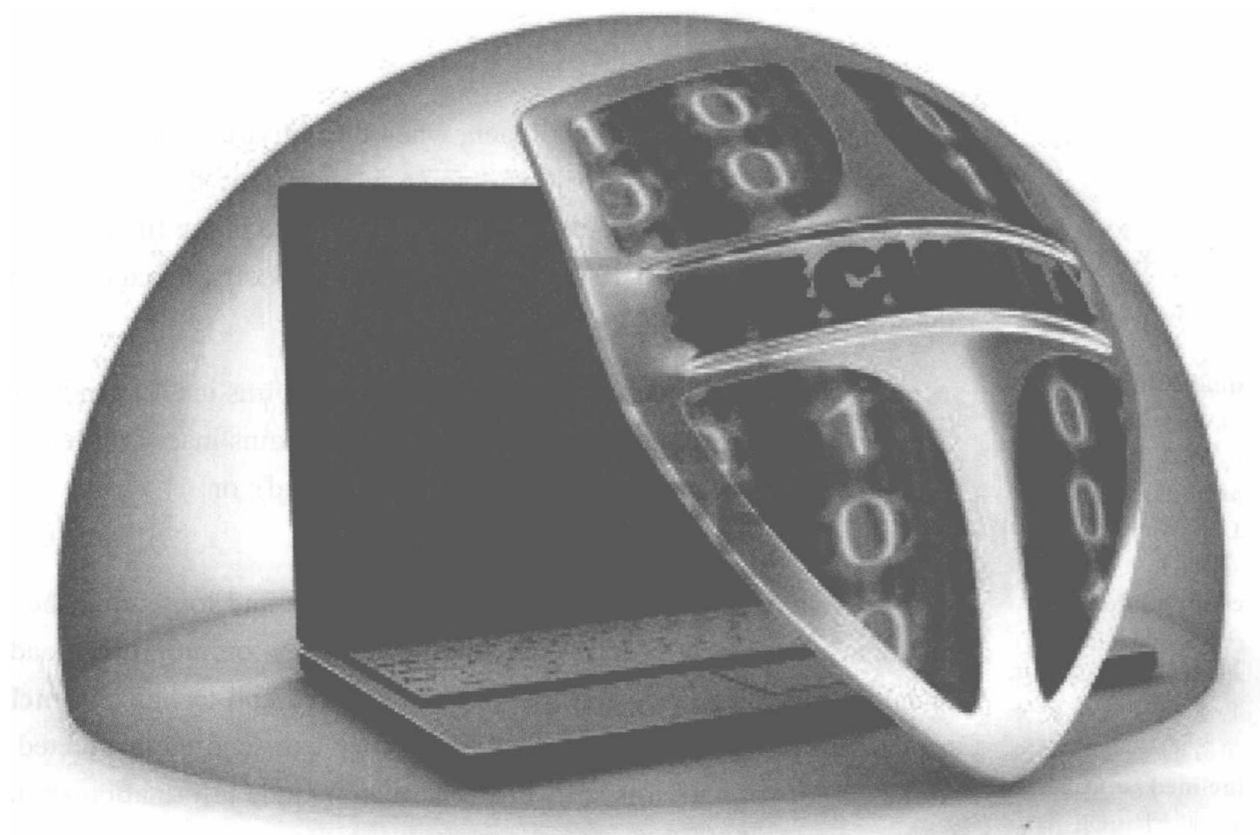
Intruder (入侵者) attacks range from the *benign* (良性的, 温和的) to the serious. At the benign end of the scale, there are many people who simply wish to explore internets and see what is out there. At the serious end are individuals who are attempting to read privileged data, perform unauthorized modifications to data, or disrupt the system.

The objective of the intruder is to gain access to a system or to increase the range of privileges accessible on a system. Most initial attacks use system or software vulnerabilities that allow a user to execute code that opens a back door into the system. Intruders can get access to a system by exploiting attacks such as *buffer overflows* (缓存溢出) on a program that runs with certain privileges.

Alternatively, the intruder attempts to acquire information that should have been protected. In some cases, this information is in the form of a user password. With knowledge of some other user's password, an intruder can log in to a system and exercise all the privileges accorded to the legitimate user.

Section B

Antivirus Software

**antivirus software**

防病毒软件

neutralize

/'nju:trəlaiz; 'nu:-/

v. 使成为无效; 抵消

worm /wɜ:m/

n. 蠕虫 (病毒)

phishing /'fɪʃɪŋ/

n. 网络钓鱼

rootkit /'ru:tkɪt/

n. 根工具包

malware /'mælwɛə/

n. 恶意软件

I. Introduction

Antivirus software is computer programs that attempt to identify, **neutralize** or eliminate malicious software. Antivirus is so named because the earliest examples were designed exclusively to combat computer viruses; however, most modern antivirus software is now designed to combat a wide range of threats, including worms, **phishing** attacks, rootkits, Trojan horses and other **malware**. Antivirus software typically uses two different approaches to accomplish this:

- Examining (scanning) files to look for known viruses matching definitions in a virus dictionary; and

suspicious

/ sə'spiʃəs /

a. 可疑的

heuristic / hju'ristik /a. 试探的,
启发式的**data capture**数据捕获,
数据收集

- Identifying **suspicious** behavior from any computer program which might indicate infection.

This second approach is called **heuristic** analysis. Such analysis may include **data captures**, port monitoring and other methods.

Most commercial antivirus software uses both of these approaches, with an emphasis on the virus dictionary approach.

II. Approaches

1. Dictionary

In the virus dictionary approach, when the antivirus software looks at a file, it refers to a dictionary of known viruses that the authors of the antivirus software have identified. If a piece of code in the file matches any virus identified in the dictionary, then the antivirus software can take one of the following actions:

- attempt to repair the file by removing the virus itself from the file;
- **quarantine** the file (such that the file remains inaccessible to other programs and its virus can no longer spread); or
- delete the infected file.

quarantine

/ 'kwɒrənti:n /

v. 隔离

periodic / ,piəri'ɒdik /a. 周期(性)的;
定期的**civic-minded**

/ 'sivik'maɪndɪd /

a. 关心公益的; 热心公民(或市民)事务的

inclined / ɪn'klaɪnd /

a. 倾向于...的; 有...爱好(或天赋)的

outbreak / 'aʊtbreɪk /

n. 爆发

oligomorphic

/ ,ɒlɪgə'mɔ:fɪk /

a. 寡形的

polymorphic

/ ,pɒlɪ'mɔ:fɪk /

a. 多(种)形(式)的, 多态的

metamorphic

/ ,metə'mɔ:fɪk /

a. 变形的

encrypt / ɪn'kript /

v. 把...加密

To achieve consistent success in the medium and long term, the virus dictionary approach requires **periodic** (generally online) downloads of updated virus dictionary entries. As **civic-minded** and **technically-inclined** users identify new viruses "in the wild"¹, they can send their infected files to the authors of antivirus software, who then include information about the new viruses in their dictionaries.

Dictionary-based antivirus software typically examines files when the computer's operating system creates, opens, closes or e-mails them. In this way, it can detect a known virus immediately upon receipt. Note too that a system administrator can typically schedule the antivirus software to examine (scan) all files on the computer's hard disk on a regular basis.

Although the dictionary approach can effectively contain virus **outbreaks** in the right circumstances, virus authors have tried to stay a step ahead of such software by writing "**oligomorphic**", "**polymorphic**" and more recently "**metamorphic**" viruses, which **encrypt** parts of themselves or otherwise modify themselves as a method of disguise, so as not to match

¹ *in the wild*: 在野生状态下。在本句中, 作者把未曾登记到病毒字典中的新病毒戏称为“野病毒”。

signature
 /'signetʃə/
 n. (人或物的) 识别标志; 鲜明特征
whitelisting
 /'hwaɪt'listɪŋ/
 n. 白名单编制
virus signature
 病毒特征代码

whitelist /'hwaɪtlist/
 n. 白名单

flag /flæg/
 v. 用标记表明

brand-new
 /'brænd'njuː/
 a. 全新的, 崭新的
desensitize
 /.di:'sensitaɪz/
 v. 使不敏感, 使降低敏感性

emulate /'emjuleɪt/
 v. 模拟, 仿真, 模仿
executable
 /'eksɪkjʊ:təbl/
 n. 可执行文件
invoke /ɪn'vəʊk/
 v. 调用; 激活

the virus's **signature** in the dictionary.

An emerging technique to deal with malware in general is **whitelisting**. Rather than looking for only known bad software, this technique prevents execution of all computer code except that which has been previously identified as trustworthy by the system administrator. By following this default denying approach, the limitations inherent in keeping **virus signatures** up to date are avoided. Additionally, computer applications that are unwanted by the system administrator are prevented from executing since they are not on the **whitelist**. Since modern enterprise organizations have large quantities of trusted applications, the limitations of adopting this technique rest with the system administrator's ability to properly inventory and maintain the whitelist of trusted applications. As such, viable implementations of this technique include tools for automating the inventory and whitelist maintenance processes.

2. Suspicious Behavior

The suspicious behavior approach, by contrast, doesn't attempt to identify known viruses, but instead monitors the behavior of all programs. If one program tries to write data to an executable program, for example, the antivirus software can **flag** this suspicious behavior, alert a user, and ask what to do.

Unlike the dictionary approach, the suspicious behavior approach therefore provides protection against **brand-new** viruses that do not yet exist in any virus dictionaries. However, it can also sound a large number of false positives,¹ and users probably become **desensitized** to all the warnings. If the user clicks "Accept" on every such warning, then the antivirus software obviously gives no benefit to that user. This problem has worsened since 1997, when much more non-malicious program designs came to modify other .exe files without regard to this false positive issue. Therefore, most modern antivirus software uses this technique less and less.

3. Other Approaches

Some antivirus software uses other types of heuristic analysis. For example, it could try to **emulate** the beginning of the code of each new **executable** that the system **invokes** before transferring control to that executable. If the program seems to use self-modifying code or otherwise

¹ it can also sound a large number of false positives: 这种方法也可能发出大量假的病毒警报。

appears as a virus (if it immediately tries to find other executables, for example), one could assume that a virus has infected the executable. However, this method could result in a lot of false positives.

sandbox

/ˈsænbɒks/

n. (供儿童在其中做堆沙游戏的) 沙池

nondeterministic

/ˌnɒndɪtəˈmɪnɪstɪk/

a. 不确定的, 非确定性的

Yet another detection method involves using a **sandbox**. A sandbox emulates the operating system and runs the executable in this simulation. After the program has terminated, software analyzes the sandbox for any changes which might indicate a virus. Because of performance issues, this type of detection normally only takes place during on-demand scans. Also this method may fail as viruses can be **nondeterministic** and result in different actions or no actions at all done when run—so it will be impossible to detect them from one run.¹

Some virus scanners can also warn a user if a file is likely to contain a virus based on the file type.

III. Issues of Concern

collusion

/kəˈluːʒən/

n. 共谋, 勾结, 串通

- The regular appearance of new malware is certainly in the financial interest of vendors of commercial antivirus software, but there is no evidence of **collusion**.

- Some antivirus software can considerably reduce performance. Users may disable the antivirus protection to overcome the performance loss, thus increasing the risk of infection. For maximum protection, the antivirus software needs to be enabled all the time—often at the cost of slower performance.

memory-resident

/ˈmeməriˈrezɪdnt/

a. 内存驻留的, 常驻内存的

- It is important to note that one should not have more than one **memory-resident** antivirus software solution installed on a single computer at any given time. Otherwise, the computer may be crippled and further damaged.

graphics card

显(示)卡

- It is sometimes necessary to temporarily disable antivirus protection when installing major updates such as Windows Service Packs² or updating **graphics card** drivers. Active antivirus protection may partially or completely prevent the installation of a major update.

¹ Also this method may fail as viruses can be nondeterministic and result in different actions or no actions at all done when run—so it will be impossible to detect them from one run.: 而且, 这种方法可能会失败, 因为病毒可能具有不确定性, 运行时导致不同的动作或根本没有动作。因此, 根据一次运行不可能检测出来。在句中, at all done 用来修饰 no actions; when run 将省略部分补全了, 应为 when they are run.

² Windows Service Packs: Windows 操作系统补丁包。

unsubscribe

/ˌʌnsəb'skraib/

v. 退订, 取消订购
(或订阅)**recourse**

/ri'kɔ:s; 'ri:kɔ:rs/

n. 追索权; 偿还请求; 求助

spyware /'spaiwɛə/

n. 间谍软件

masquerade

/ˌmæskə'reid, ˌmɑ:s-/

v. 假冒; 假扮

double-check

/'dʌbl'tʃek/

v. 复核;
从两方面查对**adware** /'ædwɛə/

n. 广告软件

- When you purchase antivirus software, the agreement may include a clause that your subscription will be automatically renewed, and your credit card automatically billed at the renewal time without your approval. For example, McAfee¹ requires one to **unsubscribe** at least 60 days before the expiration of the present subscription. In that case, the subscriber may contest the charges with the credit card issuer, but this **recourse** is likely to fail if in fact the subscriber had authorized such a “continuous payment authority”. Norton Antivirus² also has a default setting that includes the automatic renewal of your subscription.
- Some antivirus programs are actually **spyware masquerading** as antivirus software. It is best to **double-check** that the antivirus software which is being downloaded is actually a real antivirus program.
- Some commercial antivirus software programs contain **adware**. For example, the home/small business version of CA Anti-Virus 2008³ displays an advertisement for CA products whenever the desktop is unlocked after a period of inactivity.
- Most widely-accepted antivirus programs often do not detect newly-created viruses. This can be verified by making a program with destructive code in a language like C++.

IV. Effectiveness

Studies have shown that the effectiveness of antivirus software has been much reduced as compared with a few years ago, particularly against unknown or zero-day threats⁴.

magnify /'mægnifai/

v. 放大, 扩大

The problem is **magnified** by the changing intent of virus authors. Some years ago, it was obvious when a virus infection was present. The viruses of the day, written by amateurs, exhibited destructive behavior or popped-up screen messages.⁵ Modern viruses are often written by

¹ **McAfee**: McAfee 公司 (McAfee, Inc.), 一家防病毒软件与计算机安全公司, 成立于 1987 年, 总部设在美国加利福尼亚州圣克拉拉市, 产品有 McAfee VirusScan 杀毒软件 (俗称“麦咖啡杀毒软件”) 等。

² **Norton Antivirus**: 诺顿防病毒软件, 由赛门铁克公司 (Symantec Corporation) 出品。赛门铁克公司是一家计算机软件公司, 成立于 1982 年, 总部设在美国加利福尼亚州丘珀蒂诺市 (Cupertino)。

³ **CA Anti-Virus 2008**: CA 公司 (CA, Inc) 出品的防病毒软件。CA 公司是全球最大的 IT 管理软件提供商之一, 成立于 1976 年, 总部设在美国纽约。

⁴ **zero-day threats**: “零天”威胁, 即在漏洞发现的同一天就发生威胁的事件。

⁵ **The viruses of the day, written by amateurs, exhibited destructive behavior or popped-up screen messages.**: 当时的病毒是业余爱好者编写的, 它们表现出破坏性行为或显示出弹出式屏幕信息。

botnet /'bɒtnet/
n. 僵尸网络

professionals, financed by criminal organizations. It is not in their interests to make their viruses evident. Their purpose is to create **botnets** or steal information; consequently, they are often well-hidden. If an infected user has a less-than-effective antivirus product that says the computer is clean, then the virus may go undetected. Even major antivirus products sometimes fail to detect programs containing malicious behavior.

Exercises

I. Fill in the blanks with the information given in the text:

1. Most commercial antivirus software uses both the virus dictionary approach and the suspicious behavior approach with emphasis on the _____ approach.

2. An emerging technique to deal with malware in general is _____, which prevents execution of all computer code except that which has been previously identified as trustworthy by the system _____.

3. Instead of attempting to identify known viruses, the _____ approach provides protection against brand-new viruses that do not yet exist in any virus dictionaries.

4. Usually one should have only one type of _____ antivirus software installed on a single computer.

II. Translate the following terms or phrases from English into Chinese and vice versa:

- | | |
|-----------------------|------------|
| 1. phishing attack | 6. 数据捕获 |
| 2. graphics card | 7. 恶意软件 |
| 3. heuristic analysis | 8. 病毒特征代码 |
| 4. infected file | 9. 防病毒软件 |
| 5. virus dictionary | 10. 内存驻留程序 |

Section C

Types of Malicious Software

host program

主机程序, 宿主程序, 主程序

parasitic

/ˌpærə'sitik/

a. 寄生的

logic bomb

逻辑炸弹

backdoor /ˌbæk'dɔː/

n. 后门

self-contained

/ˌselfkən'teɪnd/

a. 自给的, 独立自主的

bot /bɒt/

n. 机器人; 僵尸程序 (= robot)

trigger /ˈtrɪɡə/

n. 引爆器; 触发器

inject /ɪn'dʒekt/

v. 注射; 注入; 引入

groundbreaking

/ˈgraʊndbreɪkɪŋ/

a. 开辟新天地的, 开拓性的

Malicious software can be divided into two categories: those that need a **host program**, and those that are independent. The former, referred to as **parasitic**, are essentially fragments of programs that cannot exist independently of some actual application program, utility, or system program. Viruses, **logic bombs**, and **backdoors** are examples. The latter are **self-contained** programs that can be scheduled and run by the operating system. Worms and **bot** programs are examples.

We can also differentiate between those software threats that do not replicate and those that do. The former are programs or fragments of programs that are activated by a **trigger**. Examples are logic bombs, backdoors, and bot programs. The latter consist of either a program fragment or an independent program that, when executed, may produce one or more copies of itself to be activated later on the same system or some other system. Viruses and worms are examples.

In the following, we briefly survey some of the key categories of malicious software.

I. Virus

A computer virus is a piece of software that can “infect” other programs by modifying them; the modification includes **injecting** the original program with a routine to make copies of the virus program, which can then go on to infect other programs. Computer viruses first appeared in the early 1980s, and the term itself is attributed to Fred Cohen¹ in 1983. Cohen is the author of a **groundbreaking** book on the subject.

Like its biological counterpart, a computer virus carries in its instructional code the recipe for making perfect copies of itself. The typical virus becomes embedded in a program on a computer. Then, whenever the infected computer comes into contact with an uninfected piece of software, a fresh copy of the virus passes into the new program. Thus, the infection can be spread from computer to computer by unsuspecting users who either

¹ **Fred Cohen:** 弗雷德·科恩(1957—), 美国计算机科学家, “计算机病毒之父”, 1987年在其著名论文《计算机病毒》(*Computer Viruses*)中首先提出了关于“计算机病毒”的概念。

swap / swɒp /
v. 交换

swap disks or send programs to one another over a network. In a network environment, the ability to access applications and system services on other computers provides a perfect culture for the spread of a virus¹.

A virus can do anything that other programs do. The difference is that a virus attaches itself to another program and executes secretly when the host program is run. Once a virus is executing, it can perform any function, such as erasing files and programs.

vector / 'vektə /
n. 传病媒介; 矢量
payload / 'peileud /
n. 有效载荷
dormant / 'dɔ:mənt /
a. 休眠的; 暂停活动(或作用)的
propagation
/ ,prəpe'geiʃən /
n. 传播; 繁殖; 蔓延
propagate
/ 'prəpegeit /
v. 传播; 繁殖; 蔓延

A computer virus has three parts: infection mechanism (also referred to as infection vector), trigger, and payload. During its lifetime, a typical virus goes through the following four phases: dormant phase, propagation phase, triggering phase, and execution phase.

II. Worm

viral / 'vaɪrəl /
a. 病毒(性)的;
病毒引起的
launching pad
发射台; 跳板;
起点

A worm is a program that can replicate itself and send copies from computer to computer across network connections. Upon arrival, the worm may be activated to replicate and propagate again. In addition to propagation, the worm usually performs some unwanted function. An e-mail virus has some of the characteristics of a worm because it propagates itself from system to system. However, we can still classify it as a virus because it uses a document modified to contain viral macro content and requires human action. A worm actively seeks out more machines to infect and each machine that is infected serves as an automated launching pad for attacks on other machines.

The concept of a computer worm was introduced in John Brunner's² 1975 SF³ novel *The Shockwave Rider*. The first known worm implementation was done in Xerox⁴ Palo Alto Labs⁵ in the early 1980s. It was nonmalicious search for idle systems to use to run a computationally intensive task.

Network worm programs use network connections to spread from system to system. Once active within a system, a network worm can behave as a computer virus or bacteria, or it could implant Trojan horse programs

¹ provides a perfect culture for the spread of a virus: 为传播病毒提供了理想的条件。

² John Brunner: 约翰·布伦纳(1934—1995), 英国科幻小说作家, 据认为他在1975年创作的小说《冲击波骑手》(*The Shockwave Rider*)中创造了worm(蠕虫病毒)一词并预言了计算机病毒的出现, 该书首次描写了在信息社会中计算机作为正义和邪恶双方斗争的工具的故事, 成为当年最佳畅销书之一。

³ SF: 科学幻想小说, 科幻小说(science fiction的首字母缩略)。

⁴ Xerox: (美国)施乐公司(Xerox Corporation), 1906年成立于美国纽约州罗切斯特, 总部位于康涅狄格州诺沃克。

⁵ Palo Alto Labs: (施乐公司)帕洛阿尔托实验室。

disruptive

/dis'rʌptiv/

a. 扰乱性的, 制造混乱的, 捣乱的

or perform any number of **disruptive** or destructive actions. A network worm exhibits the same characteristics as a computer virus: a dormant phase, a propagation phase, a triggering phase, and an execution phase

III. Backdoor**trapdoor** /'træpdɔ:/

n. 陷阱门, 天窗

A backdoor, also known as a **trapdoor**, is a secret entry point into a program that allows someone who is aware of the backdoor to gain access without going through the usual security access procedures. Programmers have used backdoors legitimately for many years to debug and test programs; such a backdoor is called a **maintenance hook**. This usually is done when the programmer is developing an application that has an authentication procedure, or a long setup, requiring the user to enter many different values to run the application. To debug the program, the developer may wish to gain special privileges or to avoid all the necessary setup and authentication. The programmer may also want to ensure that there is a method of activating the program should something be wrong with the authentication procedure¹ that is being built into the application. The backdoor is code that recognizes some special sequence of input or is triggered by being run from a certain **user ID**² or by an unlikely sequence of events.

maintenance hook

维护陷阱

user ID

用户标识 (符)

Backdoors become threats when **unscrupulous** programmers use them to gain unauthorized access. It is difficult to implement operating system controls for backdoors. Security measures must focus on the program development and software update activities.

unscrupulous

/ˌʌn'skru:pjʊləs/

a. 肆无忌惮的; 无耻的

IV. Logic Bomb**predate** /pri:'deɪt/

v. 在日期上早于 (或先于)

One of the oldest types of program threat, **predating** viruses and worms, is the logic bomb. The logic bomb is code embedded in some legitimate program that is set to “explode” when certain conditions are met. Examples of conditions that can be used as triggers for a logic bomb are the presence or absence of certain files, a particular day of the week or date, or a particular user running the application. Once triggered, a bomb may alter or delete data or entire files, cause a machine halt, or do some other damage.

V. Trojan Horse

A Trojan horse is a useful, or apparently useful, program or command

¹ *should something be wrong with the authentication procedure*: 万一验证过程有点毛病。should 可用于倒装句, 放在主语前面, 表示语气较强的假设, 意思相当于 if 引导的条件状语从句 (if something is wrong with the authentication procedure)。

² *ID*: 身份证明, 身份识别, 标识 (identification 的缩略)。

procedure containing hidden code that, when invoked, performs some unwanted or harmful function.

Trojan horse programs can be used to accomplish functions indirectly that an unauthorized user could not accomplish directly. For example, to gain access to the files of another user on a shared system, a user could create a Trojan horse program that, when executed, changes the invoking user's file permissions so that the files are readable by any user. The author could then induce users to run the program by placing it in a common directory and naming it such that it appears to be a useful utility program or application. An example of a Trojan horse program that would be difficult to detect is a compiler that has been modified to insert additional code into certain programs as they are compiled, such as a system login program. The code creates a backdoor in the login program that permits the author to log on to the system using a special password. This Trojan horse can never be discovered by reading the source code of the login program.

motivation

/ˌməʊti'veɪʃən/

n. 动机, 诱因

calculator program

计算(器)程序

Another common **motivation** for the Trojan horse is data destruction. The program appears to be performing a useful function (e.g., a **calculator program**), but it may also be quietly deleting the user's files.

VI. Mobile Code

Mobile code refers to programs (e.g., script, macro, or other portable instruction) that can be shipped unchanged to a heterogeneous collection of platforms and execute with identical **semantics**. The term also applies to situations involving a large homogeneous collection of platforms (e.g., Microsoft Windows).

semantics

/si'mæntiks/

n. 语义(学)

Mobile code is transmitted from a remote system to a local system and then executed on the local system without the user's explicit instruction. Mobile code often acts as a mechanism for a virus, worm, or Trojan horse to be transmitted to the user's workstation. In other cases, mobile code takes advantage of vulnerabilities to perform its own exploits, such as unauthorized data access or root compromise.¹ Popular vehicles for mobile code include Java applets, ActiveX², JavaScript, and

¹ *In other cases, mobile code takes advantage of vulnerabilities to perform its own exploits, such as unauthorized data access or root compromise.*: 在其他情况下, 移动代码利用弱点为所欲为, 如未经授权的数据访问或根权限入侵。exploit 作名词用时, 读作 /'eksplɔɪt/ (作动词用时读作 /ɪk'splɔɪt/) 表示“业绩”、“功绩”、“英勇(或令人激动、令人关注)的行为”等。

² *ActiveX*: 在广义上是指微软公司的整个 COM(组件对象模型)架构, 但是现在通常用来称呼基于标准 COM 接口来实现对象连接与嵌入的 ActiveX 控件。

attachment

/ə'tætʃmənt/

n. 附件

zombie /'zɒmbi/n. 还魂尸, 僵尸;
僵 (进程)**drone** /druːn/

n. 游手好闲者, 寄生虫

stealthy /'steɪli/

a. 偷偷摸摸的, 暗中进行的, 秘密的

registry /'redʒɪstri/

n. 注册表; 登记簿

VB-Script.¹ The most common ways of using mobile code for malicious operations on local systems are cross-site scripting², interactive and dynamic Web sites, e-mail **attachments**, and downloads from untrusted sites or of untrusted software.

VII. Bot

A bot (robot), also known as a **zombie** or **drone**, is a program that secretly takes over another Internet-attached computer and then uses that computer to launch attacks that are difficult to trace to the bot's creator. The bot is typically planted on hundreds or thousands of computers belonging to unsuspecting third parties. The collection of bots often is capable of acting in a coordinated manner; such a collection is referred to as a botnet.

A botnet exhibits three characteristics: the bot functionality, a remote control facility, and a spreading mechanism to propagate the bots and construct the botnet.

VIII. Rootkit

A rootkit is a set of programs installed on a system to maintain administrator (or root) access to that system. Root access provides access to all the functions and services of the operating system. The rootkit alters the host's standard functionality in a malicious and **stealthy** way. With root access, an attacker has complete control of the system and can add or change programs and files, monitor processes, send and receive network traffic, and get backdoor access on demand.

A rootkit can make many changes to a system to hide its existence, making it difficult for the user to determine that the rootkit is present and to identify what changes have been made. In essence, a rootkit hides by subverting the mechanisms that monitor and report on³ the processes, files, and **registries** on a computer.

IX. Multiple-Threat Malware

Viruses and other malware may operate in multiple ways. The terminology is far from uniform; this subsection gives a brief introduction to several related concepts that could be considered multiple-threat malware.

¹ *Popular vehicles for mobile code include Java applets, ActiveX, JavaScript, and VB-Script.*: 传播移动代码的流行媒介包括 Java 小应用程序、ActiveX、Java 脚本语言和 Visual Basic 脚本语言。

² *cross-site scripting*: 跨网站脚本 (攻击), 跨站脚本 (攻击)。

³ *report on*: 报告, 汇报。

multipartite

/ˌmʌltiˈpɑːtaɪt/

a. 分成多部分的

eradication

/ɪˌrædiˈkeɪʃən/

n. 根除; 消灭

contagion

/kənˈteɪdʒən/

n. (接)触(传)染

severity /sɪˈverɪti/

n. 剧烈; 严重; 严厉

A **multipartite** virus infects in multiple ways. Typically, the multipartite virus is capable of infecting multiple types of files, so that virus **eradication** must deal with all of the possible sites of infection.

A blended attack uses multiple methods of infection or transmission, to maximize the speed of **contagion** and the **severity** of the attack. Some writers characterize a blended attack as a package that includes multiple types of malware. An example of a blended attack is the Nimda attack¹, erroneously referred to as simply a worm. Nimda uses four distribution methods: e-mail, Windows shares², Web servers, and Web clients. Thus, Nimda has worm, virus, and mobile code characteristics. Blended attacks may also spread through other services, such as instant messaging and peer-to-peer file sharing.

Exercises

I. Fill in the blanks with the information given in the text:

1. A computer virus has three parts: _____ mechanism, trigger, and _____.
2. A backdoor is also known as a(n) _____.
3. A collection of bots capable of acting in a coordinated manner is referred to as a(n) _____.
4. A rootkit is a set of programs installed on a system to maintain _____ or _____ access to that system.

II. Translate the following terms or phrases from English into Chinese and vice versa:

- | | |
|-----------------------------|------------|
| 1. maintenance hook | 5. 系统登录程序 |
| 2. multipartite virus | 6. 逻辑炸弹 |
| 3. authentication procedure | 7. 多威胁恶意软件 |
| 4. instant messaging | 8. 源代码 |

¹ *Nimda attack*: 尼姆达病毒攻击。尼姆达病毒用 JavaScript 脚本语言编写, 通过电子邮件、共享网络资源、IIS (互联网信息服务) 服务器传播, 同时也是一种感染本地文件的病毒。

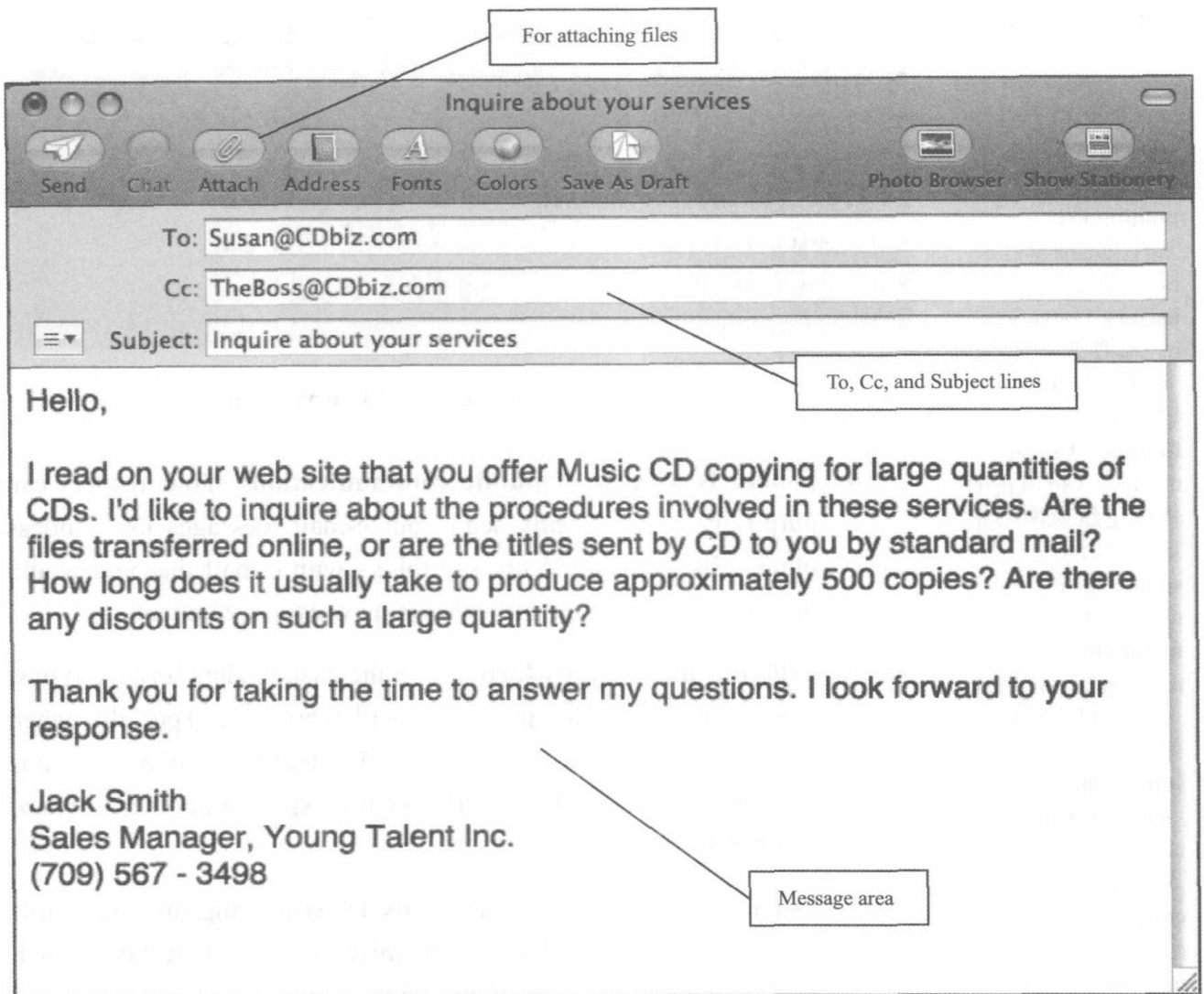
² *Windows shares*: Windows 共享文件夹。

Unit 11 Cyberculture

(计算机文化)

Section A

Using E-Mail



I. Introduction

E-mail is an electronic system for sending and receiving messages and files over a computer network. Compared to the phone or paper-based documents, e-mail is still relatively new in the workplace. When you are using e-mail, here are a few guidelines to keep in mind:

chatty /'tʃæti/

a. 聊天式的, 轻松而亲切的; 爱闲聊的

frivolous /'frɪvələs/

a. 轻薄的; 琐屑的

achiever /ə'tʃi:və/

n. 事业成功的人

signature file

签名文件

mailing list

邮件发送清单,

邮件列表

listserv /'listse:v/

n. 邮件发送清单
(或邮件列表) 管理程序

flaming /'fleɪmɪŋ/

n. 争论(特指在邮件讨论组或网络论坛中争论)

spam /spæm/

n. 垃圾邮件

netiquette /'netiket/

n. 网络礼节, 网规
(network etiquette 的缩合)

supervisor

/ˈsju:pəvaɪzə/

n. 监督(人); 管理人; 指导者

coworker

/kəu'wɜ:kə/

n. 同事

typo /'taɪpəu/

n. 打字(或排印)错误

blooper /'blu:pə/

n. 过失, 失礼

- **E-mail is increasingly used for professional purposes**—Not long ago, e-mail was considered a secondary form of communication. It was spontaneous and **chatty**, used mostly for quick comments or nonessential information. Today, e-mail is a principal form of communication in most workplaces, so people expect e-mail messages to be professional and **nonfrivolous**.

	Yes	No	Score
• Can you receive and send messages?	☐	☐	Yes Marks
• Can you forward documents to others?	☐	☐	Expert 8-10
• Can you receive and send attached documents?	☐	☐	Achiever 7-8
• Do you sort your e-mail messages into folders?	☐	☐	Beginner 5-6
• Can you send one e-mail message to multiple people?	☐	☐	Get to Work 0-5
• Have you created and used a signature file?	☐	☐	
• Are you on any mailing lists (listservs) ¹ ?	☐	☐	
• Do you know what flaming is?	☐	☐	
• Do you know what spam is?	☐	☐	
• Do you know what netiquette is?	☐	☐	

Figure 11A-1: Test Your E-Mail Readiness

- **E-mail is a form of public communication**—Your readers can purposely or mistakenly send your e-mail messages to countless others. So, you should not say things with e-mail that you would not say openly to your **supervisors**, **coworkers**, or clients.
- **E-mail is increasingly formal**—In the past, readers forgave **typos**, misspellings, and **bloopers** in e-mail messages, especially when e-mail was new and difficult to use. Today, readers expect e-mails to be more formal, reflecting the quality expected in other forms of communication.
- **E-mail standards and conventions are still being formed**—How e-mail should be used in the workplace is still being worked out. People hold widely different views about the appropriate (and inappropriate) use of e-mail. So, you need to pay close attention to how e-mail is used in your company and your readers' companies. Many companies are developing policies explaining how e-mail should be used. If your company has a policy on e-mail usage, you

¹ *mailing lists (listservs)*: 在文中指邮件讨论组(新闻组)。

should read it and follow it.

You should also keep in mind that legal constraints shape how e-mail is used in the workplace. E-mail, like any other written document, is protected by copyright law. So, you need to be careful not to use e-mails in any way that might **violate** copyright law. For example, if you receive an e-mail from a client, you cannot immediately post it to your company's **website** without that client's permission.

violate /'vaɪəleɪt/

v. 违反, 违犯

website /'websaɪt/

n. 网站, 站点

memo

/ˈmɛməʊ, ˈmi:-/

n. 〈口〉备忘录

(= memorandum)

antitrust

/ˌæntɪˈtrʌst/

a. 反托拉斯的,
反垄断的

Also, lawyers and courts treat e-mail as written communication, equivalent to a **memo** or letter. For example, much of the **antitrust** case against Microsoft in the late 1990s was built on recovered e-mail messages in which Bill Gates¹ and other executives chatted informally about aggressively competing with other companies.

Legally, any e-mail you send via the employer's computer network belongs to the employer. So, your employers are within their rights to read your e-mail without your knowledge or permission. Also, deleted e-mails can be retrieved from the company's servers, and they can be used in a legal case.

harassment

/ˈhærəsment/

he'ræs-/

n. 骚扰; 烦扰

discrimination

/dɪs.krɪmɪˈneɪʃən/

n. 区别; 歧视

hinge /hɪndʒ/

v. 依...而定, 以...
为转移 (*on, upon*)

indiscreet

/ˌɪndɪˈskri:t/

a. 不慎重的,
轻率的

orientation

/ˌɔ:riənˈteɪʃən/

n. 方向; 定向

sexual orientation

(对同性或异性的)
性取向, 性倾向

Increasingly, **harassment** and **discrimination** cases **hinge** on evidence found in e-mails. Careless e-mails about personal relationships or appearances can be saved and used against the sender in a court case. **Indiscreet** comments about gender, race, or **sexual orientation** can also have unexpected consequences. Your "harmless" dirty jokes sent to your coworkers might end up being used by a lawyer to prove that you are creating a "hostile workplace environment."

II. Basic Features of E-Mail

An e-mail is formatted similarly to a memo. Typical e-mail messages will have a *header* and *body*. They also have additional features like *attachments* and *signatures*.

1. Header

The header has lines for *To* and *Subject*. Usually, there are also lines like *cc*², *bcc*³, and *Attachments*, which allow you to expand the capabilities

¹ **Bill Gates**: 比尔·盖茨 (1955—), 美国微软公司董事长, 18岁入读哈佛大学, 读到二年级时辍学, 后来创立微软公司。

² *cc*: 抄送 (carbon copy 的首字母缩略)。

³ *bcc*: 密送 (blind carbon copy 的首字母缩略)。

of the message.

- **To line**—Here is where you type the e-mail address of the person to whom you are sending the e-mail. You can put multiple addresses on this line, allowing you to send your message to many people.
- **cc and bcc lines**—They are used to copy the message to people who are not the primary readers, like your supervisors or others who might be interested in your conversation. The cc line shows your message's recipient that others are receiving copies of the message too. The bcc line ("blind cc") allows you to copy your messages to others without anyone else knowing.
- **Subject line**—It signals the topic of the e-mail. Usually a small phrase is used. If the message is a response to a prior message, e-mail programs usually automatically insert a "Re:"¹ into the subject line. If the message is being forwarded, a "Fwd:"² is inserted in the subject line.
- **Attachments line**—It signals whether there are any additional files, pictures, or programs attached to the e-mail message. You can attach whole documents created on your **word processor**, **spreadsheet program**, or **presentation software**. An attached document retains its original formatting and can be downloaded right to the reader's computer.

word processor
文字处理软件
spreadsheet
program
电子数据表程序
presentation
software
演示软件

2. Message Area

After the header, the *message area* is where you can type your comments to your readers. It should have a clear introduction, body, and conclusion.

- The *introduction* should minimally (1) define the subject, (2) state your purpose, and (3) state your main point. Also, if you want the reader to do something, you should mention it up front³, not at the end of the e-mail.
- The *body* should provide the information needed to prove or

¹ *Re:* response 的缩写, 表示“回复”。

² *Fwd:* forward 的缩写, 表示“转发”。

³ *up front*: 〈口〉在最前面(或最突出)的位置; 直率地; 预先。

support your e-mail's purpose.

- The *conclusion* should restate the main point and look to¹ the future. Most readers of e-mail never reach the conclusion, so you should tell them any action items early in the message and then *restate* them in the conclusion.

The message area might also include these other kinds of text:

- **Reply text**—When you reply to a message, most e-mail programs allow you to copy parts of the original message into your message. These parts are often identified with > arrows running down the left margin.²
- **Links**—You can also add in direct links to websites. Most programs will automatically recognize a webpage address like “http://www.predatorconservation.org” and make it a live link in the e-mail's message area.
- **Attachments**—If you attach a file to your e-mail message, you should tell the readers in the message area that a file is attached. Otherwise, they may not notice it.

webpage

/ˈwebpeɪdʒ/

n. 网页

wink /wɪŋk/

v. 眨眼; 眨眼示意,
使眼色

grim /ɡrɪm/

a. 严厉的; 阴森的

smirk /smɜ:k/

v. 假笑; 得意地笑

Symbols	Meaning
: -)	Happy face
;-)	Winking happy face
: - o	Surprised face
: -	Grim face
: - \	Smirking face
: - (	Unhappy face
;- (	Winking unhappy face
: - x	Silent face
: - (	Crying

Figure 11A-2: Some Commonly Used Emoticons

emoticon

/iˈməʊtɪkən/

n. 情感符 (emotion
icon 的缩合)

sparing /ˈspeəriŋ/

a. 节约的;
有节制的

- **Emoticons**—Another common feature in the message area is the use of *emoticons* (Figure 11A-2). Used sparingly, they can help you signal emotions that are hard to convey in written text. When

¹ look to: 盼望, 展望。

² These parts are often identified with > arrows running down the left margin.: 这些部分常常用沿着左面的页边空白向下排列的箭头 “>” 标出。

overused, emoticons can become annoying to some readers. In most workplace situations, emoticons should not be used. They are playful and informal; therefore, they are really only appropriate in e-mails between close colleagues or friends.

3. Signature

E-mail programs usually let you create a *signature file* that automatically puts a *signature* at the end of your messages. Signature files can be both simple and complex. They allow you to personalize your message and add in additional contact information. By creating a signature file, you can avoid typing your name, title, phone number, etc. at the end of each message you write.

4. Attachments

One of the advantages of e-mail over other forms of communication is the ability to send and receive attachments. Attachments are files, pictures, or programs that readers can download to their own computer.

- **Sending attachments**—If you would like to add an attachment to your e-mail message, click on the button that says “Attach Document” or “Attachment” in your e-mail software program. Most programs will then open a **textbox** that allows you to find and select the file you want to attach.
- **Receiving attachments**—If someone sends you an attachment, your e-mail program will use an icon to signal that a file is attached to the e-mail message. Click on that icon. Most e-mail programs will then allow you to save the document to your hard disk. From there, you can open the file.

textbox /'tekstbɒks/
n. 文本(或正文、
文字)框

III. E-Mail Netiquette

E-mail has its own etiquette, or “netiquette.” Here are a dozen **netiquette guidelines** that you can follow in the workplace:

- **Be concise**—Keep the length of your messages under a screen and a half. If you have more than a screen and a half of content, pick up the phone, write a memo, or send the information as an attachment.
- **Provide only need-to-know information**—Decide who needs to know what you have to say, and send them only information they need—nothing more.

etiquette /'etiket/
n. 礼节; (行业中的)
道德规范; 规矩
concise /kən'saɪs/
a. 简明的, 简要的

critter /'krite/
n. 〈美口〉生物;
动物

irony /'aɪərəni/
n. 反语; 讽刺文体

sarcasm
/'sɑ:kæzəm/
n. 讽刺, 挖苦

witty /'witi/
a. 诙谐的;
说话风趣的

quip /kwɪp/
n. 妙语, 俏皮话

excerpt /ek'sə:pt/
v. 摘录; 引用

paraphrase
/'pærəfreɪz/
v. 将...释义(或
意译)

information
superhighway
信息高速公路

plus /plʌs/
ad. 〈口〉外加地;
另外

abusive /ə'bjʊ:sɪv/
a. 谩骂的; 毁谤的

harass /'hærəs/
v. 骚扰; 烦扰

- **Treat the security of the message about the same as a message on a postcard**—Recognize that anyone can pass your message along or use it against you. If the message is confidential or proprietary, e-mail is not an appropriate way to send it.
- **Don't say anything over e-mail that you would not say in a meeting or to legal authorities**—Hey, accidents happen. You might send something embarrassing to the entire organization. Or, someone might do it for you.
- **Never immediately respond to a message that made you angry or upset**—Give yourself time to cool off. You should never write e-mail when you are angry, because these little critters have a way of returning to bite you.
- **Avoid using too much humor, especially irony or sarcasm**—In an e-mail, your witty quips rarely come off exactly as you intended.¹ Remember that your e-mails can be easily misinterpreted or taken out of context. The American sense of humor, meanwhile, is often disturbing to international audiences; so if your message is going overseas, keep the humor to a minimum.
- **Be extra careful about excerpting or forwarding the e-mail of others**—If there is the remotest chance that someone else's message could be misunderstood, paraphrase it in your own message instead of forwarding it or copying parts of it.
- **Don't be unpleasant over e-mail if you would not be so face to face**—E-mail provides a false sense of security, much like driving in a car. But in reality, those are real people on that information superhighway,² and they have feelings. Plus they can save your abusive or harassing messages and use them as evidence against you.
- **Never send anything that could be proprietary**—There have

¹ *In an e-mail, your witty quips rarely come off exactly as you intended.*: 在电子邮件中, 你的诙谐俏皮话难得达到你想要的确切效果。come off 表示“以某种方式结束”、“获得某种结果”、“结果是”等。

² *those are real people on that information superhighway*: 信息高速公路上的那些人都是真人。句中的 those 与其定语 (on that information superhighway) 被谓语 (系表结构) 分隔开。如果把 on that information superhighway 作为 people 的定语, 那么 those 是指哪些人则不好解释, 因为上下文没有提到。这实际上是一个分隔结构。分隔结构指语法关系密切的两个句子成分被其他句子成分分隔开的现象。分隔结构的产生多是为了保持句子平衡、避免头重脚轻, 或是为了语义严密、结构紧凑等。

classified

/'klæsɪfaɪd/

a. 归入密级的,
保密的**courier** /'kʊəriə/

n. 信使

flame /fleɪm/v. (向...) 发送争论
(或争辩) 邮件**spam** /spæm/v. (向...) 发送垃圾
邮件**chain-letter**

/'tʃeɪn.leɪtə/

v. 向...发送连锁信
(或连锁邮件)**hoax** /həʊks/

n. 骗局; 恶作剧

gullible /'ɡʌlibəl/a. 易受骗的, 易上
当的**sloppy** /'slɒpi/a. (口) 马虎的;
凌乱的

been some classic cases where classified information or trade secrets have been sent to competitors or the media over e-mail. If the information is proprietary, use traditional routes like the regular mail or a courier.

- **Don't flame, spam, or chain-letter¹ people at work**—If you must, do these things from your personal computer at home. At work, assume these activities will ultimately be used against you. You might be fired.
- **Think twice before you send “urgent” messages that are making their way across the net**—There are lots of hoaxes about computer viruses “erasing your hard drive.” These hoaxes gain new life when some gullible persons start sending them to all their friends. Some viruses are real, but most are not. Wait a day or two to see if the threat is real before you warn all your friends and colleagues.
- **Be forgiving of the grammatical mistakes of others—but don't make them yourself**—Some people treat e-mail informally, so they rarely revise (though they should). Grammar mistakes and misspellings happen—so you can forgive the sender. On the other hand, these mistakes and typos make the sender look stupid. Don't send out sloppy e-mail messages.

Exercises

I. Fill in the blanks with the information given in the text:

1. As used in the text, e-mail can refer both to a single electronic message and to the entire electronic _____ for sending and receiving messages and files over a computer network.
2. E-mail is protected by _____ law. If you receive an e-mail from a client, you cannot immediately post it to your company's website without his or her permission.
3. Legally, e-mail messages sent by employees via the employer's

¹ **chain-letter**: 源自名词 chain letter, 即“连锁信”(分别寄给数人的信, 并要求他们复制此信若干份, 再分别寄出)。

computer network belong to the employer, so the employer is within his _____ to read them without the employees' knowledge or permission.

4. When you are using e-mail, you should remember that _____ e-mail messages can be retrieved from your company's servers and used in a legal case.

5. An attached document retains its original _____ when it is received by the reader and can be downloaded right to his or her computer.

6. A(n) _____ file created with e-mail programs can automatically put a signature at the end of each e-mail message.

7. E-mail messages should be concise. Usually, their length should be kept under one and a half _____.

8. Because _____ are playful and informal, they are only appropriate in e-mails between close colleagues or friends and should not be used in most workplace situations.

II. Translate the following terms or phrases from English into Chinese and vice versa:

- | | |
|------------------------------|-------------|
| 1. mailing list | 11. 常用情感符 |
| 2. proprietary software | 12. 已删除电子邮件 |
| 3. cc line | 13. 电子系统 |
| 4. bcc line | 14. 附件行 |
| 5. forwarded e-mail messages | 15. 版权法 |
| 6. e-mail convention | 16. 电子邮件网规 |
| 7. click on an icon | 17. 信息高速公路 |
| 8. confidential document | 18. 签名文件 |
| 9. classified information | 19. 电子数据表程序 |
| 10. recovered e-mail message | 20. 文字处理软件 |

III. Fill in each of the blanks with one of the words given in the following list, making changes if necessary:

- | | | | |
|---------------|------------------|----------------|-----------------|
| <i>access</i> | <i>in-person</i> | <i>respond</i> | <i>download</i> |
| <i>create</i> | <i>see</i> | <i>cost</i> | <i>window</i> |
| <i>e-mail</i> | <i>touch</i> | <i>message</i> | <i>audio</i> |
| <i>camera</i> | <i>website</i> | <i>program</i> | <i>computer</i> |

Instant messaging has always been a good way to stay in _____ with friends. As companies look to cut _____ and improve efficiency, instant messaging is a good way to replace _____ meetings, while allowing people to *collaborate* (合作) through their _____. Instant messaging is a valuable new tool that is becoming as common as _____ in the workplace.

Several companies, such as Yahoo!, AOL, and Microsoft, will give you free _____ to instant messaging. To use their systems, you will need to _____ their software to your computer from their _____. Some instant messaging programs include the ability to send live _____ or video. With a microphone and/or digital _____ mounted on your computer, you can let other people _____ and hear you talking.

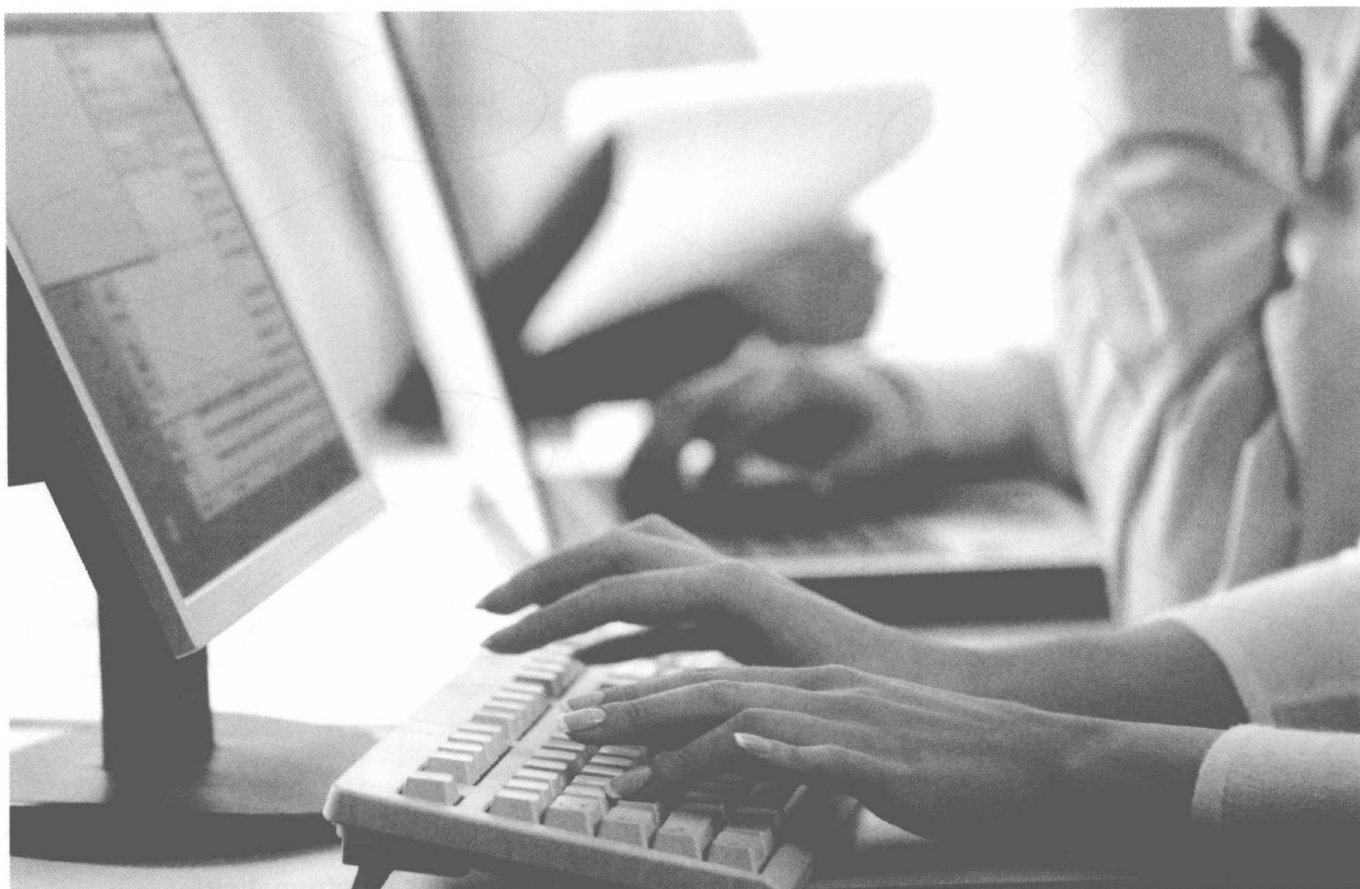
Most instant messaging _____ ask you to sign in with a password. Then, you can _____ a list of other people with whom you want to *converse* (交谈). To begin a conversation, send a _____ to the others to see if they are at their computers. Their computers will open a _____ or make a sound to tell them you want to talk to them. If they _____, you can start writing back and forth.

IV. Translate the following passage from English into Chinese:

The pace of change brought about by new technologies has had a significant effect on the way people live, work, and play worldwide. New and emerging technologies challenge the traditional process of teaching and learning, and the way education is managed. It is frequently claimed that the most *potent* (强有力的) agent of change on present-day society will prove to be information technology (IT). IT, while an important area of study in its own right, is having a major impact across all curriculum areas. Easy worldwide communication provides instant access to a vast array of data. Rapid communication, plus increased access to IT in the home, at work, and in educational establishments, could mean that learning becomes a truly lifelong activity—an activity in which the pace of technological change forces constant evaluation of the learning process itself.

Section B

Designing Websites



I. Organizing and Drafting a Website

Organizing and drafting a website is not all that¹ different from organizing and drafting a print document.

1. Organizing the Website

Professional website developers often like to start creating a new site by **mapping** out its contents on a **whiteboard**, piece of paper, or screen. Logical mapping is a good way to sort out the content of the website and develop an efficient organizational scheme for it.

To map out the contents of the site, start by writing your website's purpose in the middle of your screen or a sheet of paper. Then, use lines and

map /mæp/
v. 详细安排, 筹划
(*out*)
whiteboard
/'hwaɪtbɔ:d/
n. 白色书写板

¹ *all that*: [用于否定句] (并不是) 那么, 如此。

circles to begin identifying the contents of the site. (Figure 11B-1)

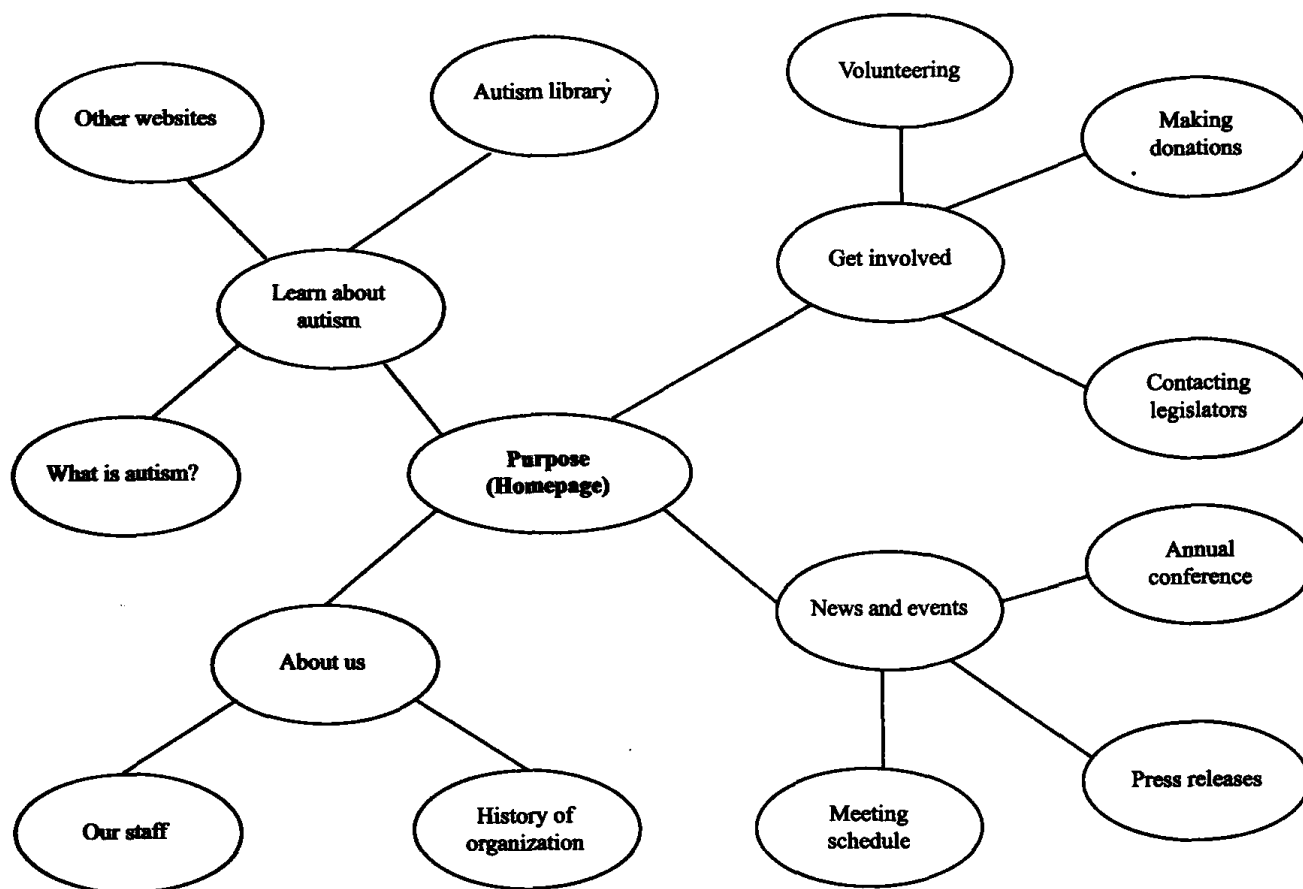


Figure 11B-1: Mapping out a website is a good way to see the visual relationships among the site's pieces of information.

low-tech /,ləu'tek/
a. 低技术的

You might also try a **low-tech** method that is popular with professional website designers—using sticky notes. On a blank wall, use sticky notes to map out your website. This low-tech method allows you and your team to move the notes around to look at different ways to organize the site. Another advantage of sticky notes is that they can be added or eliminated with ease.

2. Creating Levels in the Website

Your logical map and research will likely give you a sense of how many levels are needed in the website. Professional website designers use the following guidelines to determine the number of levels in a website:

- a maximum of three links for the most important information.
- a maximum of five links for 80 percent of all information.

- a maximum of seven links for all information.

These guidelines are helpful, because if you force your readers, especially customers, to **wade** through too many pages, you risk losing them.

wade / weɪd /
v. 跋涉,
艰难地行走

3. Drafting the Pages of the Website

Websites have a few different kinds of pages.

homepage
/ 'həʊmpeɪdʒ /
n. 主页

(1) Homepage

The homepage is usually the first page of a website. In many ways, it is like an introduction and table of contents for the website. It identifies the subject and purpose of the site and usually includes a **navigation bar** that forecasts the overall structure of the site. The navigation bar can appear just about anywhere, but it typically appears along the top of the page or the left-hand side. Keep the homepage as simple as possible, saving the main content of the site for the pages that follow.

navigation bar
导航条

node page
节点页面

(2) Node Pages

Node pages typically follow links on the homepage, further dividing the subject of the site. For example, a university's homepage will have links that go to node pages like *Academics*, *Colleges and Departments*, *Libraries*, *Students*, and *Faculty and Staff*. Each node page introduces readers to one of the site's major topics. A well-designed node page allows readers to select among other further topics. In essence, it serves as a **miniature** homepage to this part of the site.

miniature / 'mɪniətʃə;
'mɪniətʃər/
a. 小型的, 微小的

(3) Basic Pages

Basic pages provide the content (e.g., facts, data, examples, details, descriptions) that readers are looking for. In a basic page, you should keep the discussion limited to a topic that can be handled in about a screen and a half. Readers of websites rarely have enough patience to scroll down farther. So, if you need more than two screens to cover a particular issue, you might need two or more basic pages.

navigation page
导航页面

(4) Navigation Pages

site map
网站地图

Navigation pages help readers find the information they are seeking. These pages include a **site map** that lists all the topics covered in a website, like an index in a book. Larger websites may also include an internal search engine. Search engines are available, usually free, from the larger search

engines like Excite¹ (excite.com), Lycos² (lycos.com), and Google (google.com). However, putting a search engine into your website takes a little more programming skill. Besides, software programs like RoboHelp³, DocToHelp⁴, and Help & Manual⁵ will allow you to create Help areas for your website. Usually, only the largest websites include a Help area.

splash page

醒目页面, 过渡页,
欢迎页面

decorative

/ˈdekerətiv/

a. 装饰性的

(5) Splash Page

This optional page comes before the homepage and acts like a cover to a book. Splash pages are mostly decorative, using animation or images to set a specific tone before readers enter the website through the homepage.⁶

A well-designed website gives readers a few different ways to access information. Preferably, they can start out at the homepage and work their way down to the information they need. If they are having trouble locating information, perhaps they can use a site map to find it. And, as a last resort, the search engine might help them find pages that discuss the topic they are looking for.

II. Using Style in a Website

Here are some strategies for improving the readability of your website.

- **Links should reflect titles**—When readers click on a link, the title of the page that follows should be the same as the link they selected. When readers click on a link saying “Research on Whales,” “Research on Whales” should be the title of the next page. Otherwise, readers may be confused for a moment.
- **Keep sentences short**—On average, sentences in websites should

¹ *Excite*: 著名的搜索网站, 提供“智能性”搜索引擎, 具有类目、网站、全文和新闻检索功能及高级检索功能, 支持逻辑条件限制查询。除美国站点外, 还有多个站点, 支持多种文字的关键字查询。

² *Lycos*: 著名的搜索网站, 提供类目、网站、图像及声音文件等多种检索功能, 并具有高级检索功能, 支持逻辑条件限制查询。搜索图像和声音文件的功能尤其强。

³ *RoboHelp*: Adobe 公司 (Adobe Systems Incorporated) 出品的一种软件, 使技术通信人员、帮助文档作者和 Web 开发人员能够为帮助系统和独立的知识库构建、管理和发布内容。Adobe 公司成立于 1982 年, 总部设在美国加利福尼亚州圣何塞市, 系一家规模庞大、多元化的软件公司。

⁴ *DocToHelp*: ComponentOne 公司出品的一种软件, 在开发 Windows 应用程序和基于 Web 的应用程序时, 能够快速容易地使用一个源文件创建和开发附加的帮助文件的打印文档。ComponentOne 公司总部设在美国宾夕法尼亚州匹兹堡市。

⁵ *Help & Manual*: EC 软件公司 (EC Software GmbH) 出品的一种用来制作专业的使用手册和帮助文件的软件。EC 软件公司设在奥地利萨尔茨堡附近, 成立于 1993 年。

⁶ *Splash pages are mostly decorative, using animation or images to set a specific tone before readers enter the website through the homepage.*: 醒目页面多半是装饰性的, 它们使用动画或图像在读者通过主页进入网站之前设定一种特定的格调。

be shorter than sentences in paper-based documents. Website readers usually scan faster than readers of paper documents.

- **Keep paragraphs short**—Paragraphs should be kept to a few sentences or less. That way, readers can scan the paragraph in a glance. If you find yourself writing a long paragraph, you might be offering too much support for your point. If so, you may need to make a separate webpage for this supporting information.
- **Use mapping to develop themes**—Since websites are more modular than paper-based documents, it is a good idea to use themes to set an overall tone that will keep the website sounding consistent. To create a theme, put the word that best represents the theme or tone you desire in the center of your screen or a sheet of paper. Then, use logical mapping to find words that are associated with that theme or tone (Figure 11B-2). As with paper-based documents, use these words **strategically** throughout the site. When your readers experience these words, they will **subconsciously** sense the theme or tone you are trying to create.

modular /'mɒdjʊlə/

a. 模块化的

strategically

/strə'ti:dʒikəli/

ad. 在战略上;
策略地

subconsciously

/ˌsʌb'kɒŋfəsli/

ad. 下意识地, 潜意识地

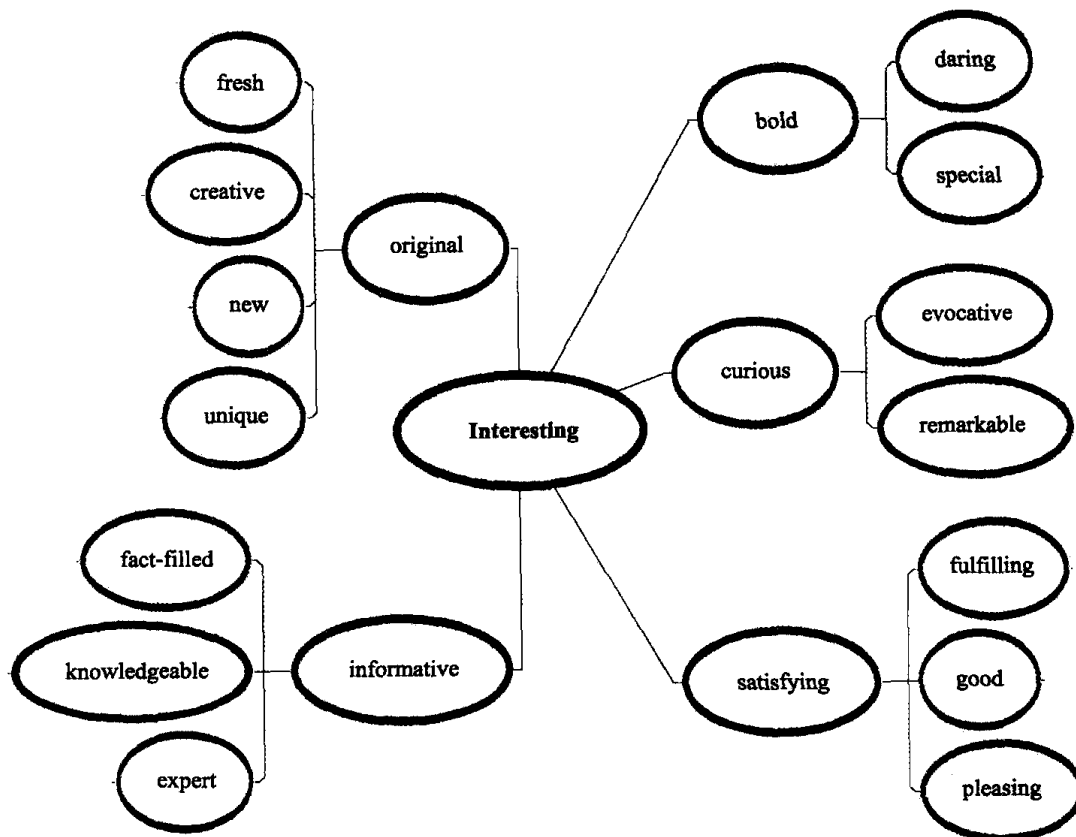


Figure 11B-2: Logical mapping can help you create themes or tones that hold the website together.

A website that uses this map would set the theme “interesting.”

III. Designing a Website

1. Designing the Interface

offset /'ɒfset; 'ɒ:f-/
v. (出于比较等目的)把...并列, 衬托出; 补偿; 抵消

alignment

/ə'laɪnmənt/

n. 对准, 对齐

align /ə'laɪn/

v. 对准, 对齐

The interface design determines in great part whether your website will be attractive to readers and whether it will be easy to use. The following are five principles of design for the interface of a website:

- **Balance**—Items placed on the screen **offset** each other to create a feeling of stability on the page or screen.
- **Alignment**—**Aligned** items on the page help readers identify different levels of information in the interface.

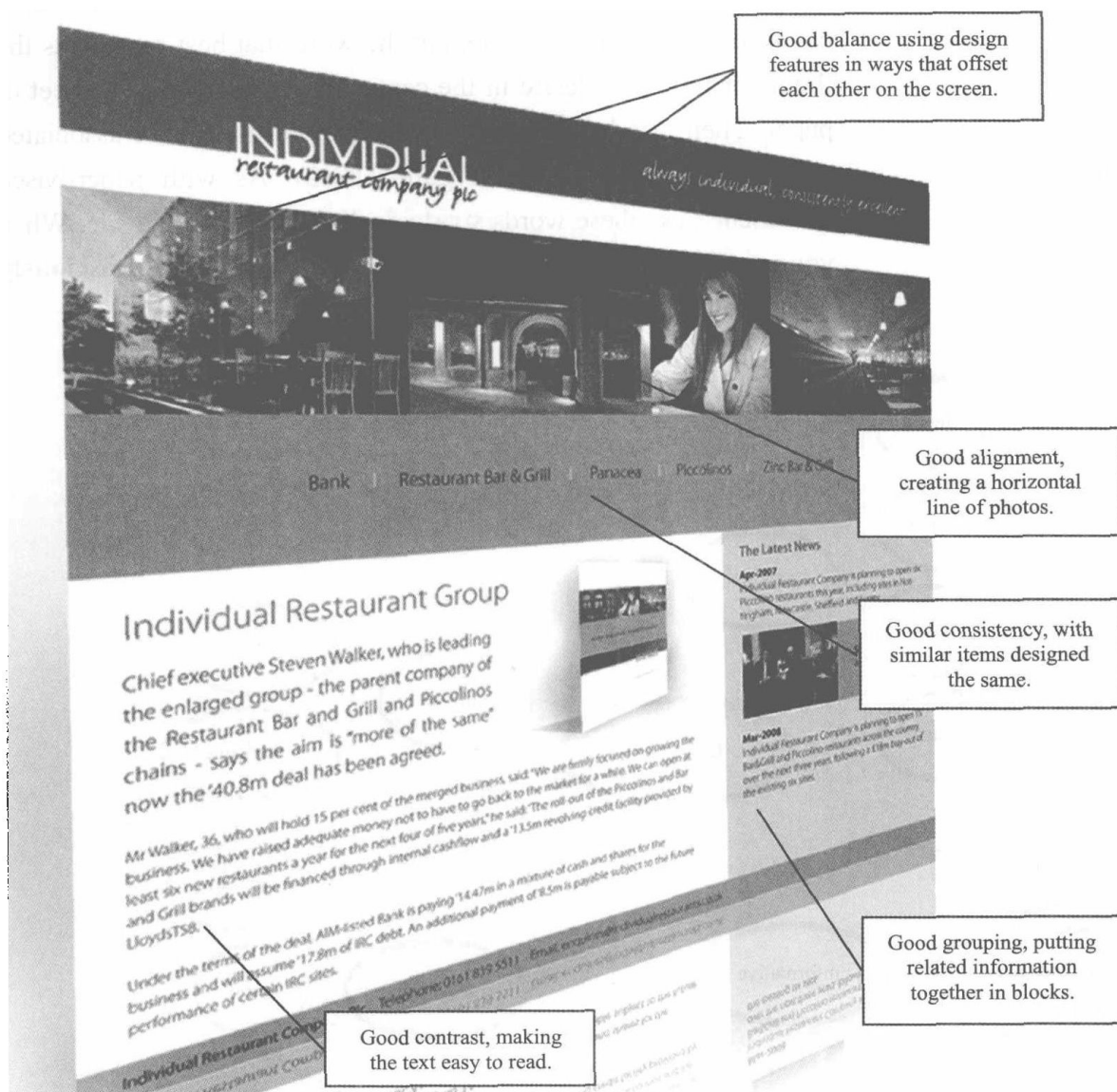


Figure 11B-3: This homepage uses all the design principles well.

- **Grouping**—Items that are near each other on the screen will be seen as being “grouped.”

- **Consistency**—Each page in the website looks similar to the others, minimizing chaos in the interface.
- **Contrast**—Contrast sharpens the relationships between images and text, headings and text, and text and the background.

These principles should be used to develop a consistent interface, called a template, for your webpages. The template will give your website a standardized, predictable design, allowing your readers to quickly locate the information they are seeking.

2. Adding Images

Images in websites are saved as separate files on your computer and on the server. These files include any pictures, graphs, or drawings you might want to include on your webpage. They also include elements of the interface, such as banners on the screens, corporate logos, and icons.¹ Images can be saved in a variety of file formats. The two formats most commonly used for websites are called *jpeg*² and *gif*³ formats.

To put images in your webpages, you need to do three things.

- **Create or capture the image.** You can use a digital camera, scanner, or drawing software to create an image. Then, you can use a program like Adobe Photoshop⁴ to turn the file into a jpeg or gif. Or, you might find an image on a website or CD-ROM and download it to your hard drive.
- **Place the image.** Put the image file in the folder on your computer that holds the other files for your website. When possible, keep images with the pages that reference them. Otherwise, there is a chance the image will not be transferred with the page, causing the image not to be shown on the page.
- **Insert the image.** Using your web-authoring software, open the page in which the image will appear. Then, use the Insert Image command (or equivalent) to add the image to the page. When you are finished inserting the image, it should appear on the webpage.

banner /'bænə/
n. 标题, 横幅
logo /'ləʊɡəʊ/
n. 标识, 标志图, 徽标

author /'ɔ:θə/
v. 著作, 写作;
编写
web-authoring software
网络写作软件

¹ *such as banners on the screens, corporate logos, and icons*: 如屏幕上的横幅标题、公司标识和图标。

² *jpeg*: 联合图像专家小组 (Joint Photographic Experts Group 的首字母缩略), 这里指由该小组创立的图像格式。读作 /'dʒeɪpeg/。

³ *gif*: 图形交换格式 (Graphics Interchange Format 的首字母缩略), CompuServe 公司 1987 年推出的一种图像格式。

⁴ *Adobe Photoshop*: Adobe 公司出品的一款著名的图像处理软件。

proofreading

/'pru:fri:diŋ/

n. 校对

IV. Revising, Editing, and Proofreading

Revising and editing are not simply ways to polish the website. You need to look closely at the content, organization, style, and design to see if they are appropriate for your readers and the context in which the site will be used. When you are revising the website, ask yourself these questions:

Does the website highlight information my readers want?

Are the subject and purpose of the website obvious on the homepage?

Is the site organized in a way that is logical to my site's readers?

Is the style of the website appropriate? Does it set the right tone?

Is the interface easy to navigate?

Is the interface attractive and interesting?

proofread

/'pru:fri:d/

v. 校对

Proofreading a website is especially difficult, because seeing errors on the screen is often harder than seeing them on paper. So, you might print out the screens of your website and proofread them on paper.

When you have finished revising and editing the website, do some usability testing to ensure quality. Minimally, you should ask your coworkers and friends to try using the site, most likely on your computer. Depending on the importance of the site, you might also set up formal usability tests with sample readers. You can do read-and-locate tests, understandability tests, performance tests, and even safety tests.

V. Uploading, Testing, and Maintaining Your Website

To make your website available to the public, you need to upload it to a server—a large computer that connects your computer to the Internet. Before uploading to the server, most writers of websites prefer to create a finished website on their own computer. Then, they use an FTP software program to copy the whole website to the server at once. When you copy your webpage files, don't forget to also copy your image files with them.

Once your website is on the server, it is time to test it. Using at least a couple different browsers (e.g., Netscape, Explorer), look over the pages on the site. Try out the links and resize the browser window a few different ways. Usually, you will find some formatting problems that did not show up in your web-authoring program's version of the site. Another common problem is that links to other pages don't work, giving readers error messages when links to these pages are chosen. To fix these problems, reopen the files with your web-authoring program. Make the corrections

and changes needed. Then, re-upload your revised pages to the website.

Few websites can be left alone for long. You should plan to regularly update your website, usually weekly or monthly. To update a page, go back to the file saved on your hard drive and open it in your web-authoring program. Make the changes you want and then use FTP to copy the updated file onto the server. The server will replace the old file with your new file. Most companies hire or contract a webmaster to maintain their site. The webmaster regularly updates the site, incorporating any changes and eliminating any dated materials. When you are your own site's webmaster, though, you should revisit your site regularly, looking for items to update.

webmaster
/'web,mɑ:stə/
n. 网站管理员
dated /'deɪtɪd/
a. 过时的

Exercises

I. Fill in the blanks with the information given in the text:

1. When professional website designers determine the number of _____ in a website, a guideline usually followed is that there should be a maximum of _____ links for the most important information.
2. A website has a few different kinds of webpages: the homepage, _____ pages, basic pages, _____ pages, and the optional splash page.
3. According to the text, the five principles of design for the interface of a website are _____, alignment, grouping, consistency, and _____.
4. Before uploading your website, you should do some _____ testing to ensure quality.

II. Translate the following terms or phrases from English into Chinese and vice versa:

- | | |
|---------------------------|------------|
| 1. web-authoring software | 6. 导航条 |
| 2. template generator | 7. 节点页面 |
| 3. navigation page | 8. 网站地图 |
| 4. corporate logo | 9. 可用性测试 |
| 5. splash page | 10. 图形交换格式 |

Section C

Social Issues of Computer Networks

printing press
印刷机

ethical /'eθikəl/
a. 道德的; 伦理的

message board
留言板, 消息板

live-and-let-live
a. 自己活也让别人活的; 互相宽容的; 互不相扰的
verbal /'və:bəl/
a. 用言辞的; 文字上的; 口头的

pornography
/pɔ:'nɒgrəfi/
n. [总称] 色情(或淫秽)作品

censor /'sensə/
v. 审查, 检查

rage /reɪdʒ/
v. 发怒; 激烈进行

sue /sjuː; suː/
v. 控告, 起诉

police /pə'liːs/
v. 维持...的治安; 管理; 监督

Computer networks, like the **printing press** 500 years ago, allow ordinary citizens to distribute and view content in ways that were not previously possible. But along with the good comes the bad, as this new-found freedom brings with it many unsolved social, political, and **ethical** issues. Let us just briefly mention a few of them; a thorough study would require a full book, at least.

Social networks, **message boards**, content sharing sites, and a host of other applications allow people to share their views with like-minded individuals. As long as the subjects are restricted to technical topics or hobbies like gardening, not too many problems will arise.

The trouble comes with topics that people actually care about, like politics, religion, or sex. Views that are publicly posted may be deeply offensive to some people. Worse yet, they may not be politically correct. Furthermore, opinions need not be limited to text; high-resolution color photographs and video clips are easily shared over computer networks. Some people take a **live-and-let-live** view, but others feel that posting certain material (e.g., **verbal** attacks on particular countries or religions, **pornography**, etc.) is simply unacceptable and that such content must be **censored**. Different countries have different and conflicting laws in this area. Thus, the debate rages.

In the past, people have sued network operators, claiming that they are responsible for the contents of what they carry, just as newspapers and magazines are. The inevitable response is that a network is like a telephone company or the post office and cannot be expected to police what its users say.

It should now come only as a slight surprise to learn that some network operators block content for their own reasons. Some users of peer-to-peer applications had their network service cut off because the network operators did not find it profitable to carry the large amounts of traffic sent by those applications. Those same operators would probably like to treat different companies differently. If you are a big company and pay well then you get

small-time

/'smɔ:l'taɪm/

a. 〈口〉次要的, 无关紧要的

neutrality

/nju:'træliiti/

n. 中立

tussle /'tʌsəl/

n. 争执, 争辩

pirate /'paɪəret/

v. 剽窃; 盗用; 非法翻印

infringe /'ɪnfrɪndʒ/

v. 违犯, 侵犯(权利等)

millennium

/mi'leniəm/

n. 一千年; 千禧年

arms race

军备竞赛

infringement

/ɪn'frɪndʒmənt/

n. (对他人权利等的) 侵犯, 侵害

culprit /'kʌlprɪt/

n. 罪犯; 被控犯罪的人

snoop /snu:p/

v. 窥探; 打探

outgoing

/ˈaʊtɡəʊɪŋ/

a. 往外去的, 离去的

carnivore

/'kɑ:nɪvɔ:/

n. 食肉动物

good service, but if you are a **small-time** player, you get poor service. Opponents of this practice argue that peer-to-peer and other content should be treated in the same way because they are all just bits to the network. This argument for communications that are not differentiated by their content or source or who is providing the content is known as network **neutrality**. It is probably safe to say that this debate will go on for a while.

Many other parties are involved in the **tussle** over content. For instance, **pirated** music and movies fueled the massive growth of peer-to-peer networks, which did not please the copyright holders, who have threatened (and sometimes taken) legal action. There are now automated systems that search peer-to-peer networks and fire off¹ warnings to network operators and users who are suspected of **infringing** copyright. In the United States, these warnings are known as DMCA² takedown notices after the Digital Millennium Copyright Act.³ This search is an arms race because it is hard to reliably catch copyright infringement. Even your printer might be mistaken for a **culprit**.

Computer networks make it very easy to communicate. They also make it easy for the people who run the network to **snoop** on the traffic. This sets up⁴ conflicts over issues such as employee rights versus employer rights. Many people read and write email at work. Many employers have claimed the right to read and possibly censor employee messages, including messages sent from a home computer outside working hours. Not all employees agree with this, especially the latter part.

Another conflict is centered around government versus citizen's rights. The FBI⁵ has installed systems at many Internet service providers to **snoop** on all incoming and outgoing email for nuggets of interest⁶. One early system was originally called Carnivore, but bad publicity caused it to be renamed to the more innocent-sounding DCS1000⁷. The goal of such systems is to spy on millions of people in the hope of perhaps finding information about illegal activities. Unfortunately for the spies, the Fourth

¹ *fire off*: 恶狠狠地发出; 迅速发出。

² *DMCA*: 《数字千年版权法》, 《千禧年数字版权法》(Digital Millennium Copyright Act 的首字母缩略)。

³ *In the United States, these warnings are known as DMCA takedown notices after the Digital Millennium Copyright Act.*: 在美国, 这些警告按照《数字千年版权法》称为 DMCA 取下通知。

⁴ *set up*: 引起, 产生。

⁵ *FBI*: (美国) 联邦调查局 (Federal Bureau of Investigation 的首字母缩略)。

⁶ *to snoop on all incoming and outgoing email for nuggets of interest*: 窥探所有的来往电子邮件, 寻找感兴趣的信息。

⁷ *DCS*: 数字收集系统 (Digital Collection System 的首字母缩略)。

amendment

/ə'mendmənt/

n. 修正案,
修正条款**warrant**

/'wɒrənt; 'wɔ:-/

n. 授权(令)

monopoly

/mə'nɒpəli/

n. 垄断; 独有

cookie /'kuki/n. 小甜饼(指一种临时保存网络用户信息
的结构)**frequent** /fri'kwent/v. 常到, 常去, 时
常出入于**anonymous**

/ə'nɒnɪməs/

a. 匿名的

reprisal /ri'praɪzəl/

n. 报复

accusation

/ˌækju:'zeɪʃən/

n. 指控, 控告

downright

/'daʊnraɪt/

ad. 彻底地, 完全地

pluck /plʌk/

v. 拔; 摘, 采

junk /dʒʌŋk/n. 废旧物品; 破烂,
垃圾**spammer** /'spæmə/

n. 垃圾邮件发送者

Amendment to the U.S. Constitution prohibits government searches without a search **warrant**, but the government often ignores it.

Of course, the government does not have a **monopoly** on threatening people's privacy. The private sector does its bit too by profiling users.¹ For example, small files called **cookies** that Web browsers store on users' computers allow companies to track users' activities in cyberspace and may also allow credit card numbers, social security numbers, and other confidential information to leak all over the Internet. Companies that provide Web-based services may maintain large amounts of personal information about their users that allows them to study user activities directly. For example, Google can read your email and show you advertisements based on your interests if you use its email service, Gmail².

A new twist with mobile devices is location privacy.³ As part of the process of providing service to your mobile device the network operators learn where you are at different times of day. This allows them to track your movements. They may know which nightclub you **frequent** and which medical center you visit.

Computer networks also offer the potential to increase privacy by sending **anonymous** messages. In some situations, this capability may be desirable. Beyond preventing companies from learning your habits, it provides, for example, a way for students, soldiers, employees, and citizens to blow the whistle on⁴ illegal behavior on the part of professors, officers, superiors, and politicians without fear of **reprisals**. On the other hand, in the United States and most other democracies, the law specifically permits an accused person the right to confront and challenge his accuser in court so **anonymous accusations** cannot be used as evidence.

The Internet makes it possible to find information quickly, but a great deal of it is ill considered, misleading, or **downright** wrong. That medical advice you **plucked** from the Internet about the pain in your chest may have come from a Nobel Prize winner or from a high-school dropout.

Other information is frequently unwanted. Electronic **junk** mail (spam) has become a part of life because **spammers** have collected millions of

¹ *The private sector does its bit too by profiling users.*: 私营部门通过用户资料挖掘(或用户分析)在这方面也有份。do one's bit 意为“做自己应做的一份工作”,“尽本分”。

² *Gmail*: Google 提供的电子邮件服务。

³ *A new twist with mobile devices is location privacy.*: 移动设备的一个新花样是位置隐私。

⁴ *to blow the whistle (on)*: 告发, 揭发。

email addresses and would-be marketers can cheaply send computer-generated messages to them. Fortunately, filtering software is able to read and discard the spam generated by other computers, with lesser or greater degrees of success.

Still other content is intended for criminal behavior. Web pages and email messages containing active content (basically, programs or macros that execute on the receiver's machine) can contain viruses that take over your computer. They might be used to steal your bank account passwords, or to have your computer send spam as part of a botnet or pool of compromised machines.

Phishing messages masquerade as originating from a trustworthy party, for example, your bank, to try to trick you into revealing sensitive information, for example, credit card numbers. Identity theft is becoming a serious problem as thieves collect enough information about a victim to obtain credit cards and other documents in the victim's name.

It can be difficult to prevent computers from impersonating people on the Internet. This problem has led to the development of CAPTCHAs¹, in which a computer asks a person to solve a short recognition task, for example, typing in the letters shown in a **distorted** image, to show that he is human. This process is a variation on the famous Turing test² in which a person asks questions over a network to judge whether the entity responding is human.

A lot of these problems could be solved if the computer industry took computer security seriously. If all messages were encrypted and **authenticated**, it would be harder to commit **mischief**. Such technology is well established. The problem is that hardware and software vendors know that putting in security features costs money and their customers are not demanding such features. In addition, a substantial number of the problems are caused by buggy software, which occurs because vendors keep adding more and more features to their programs, which inevitably means more code and thus more bugs. A tax on new features might help, but that might be a tough sell in some quarters.³ A **refund** for **defective** software might be

distorted /dis'tɔ:tɪd/
a. 扭曲的, 变形的

authenticate
/ɔ:'θentikeɪt/
v. 验证, 鉴别
mischief /'mɪʃɪf/
n. 恶作剧; 淘气;
损害
refund /'ri:fʌnd/
n. 退款; 退还
defective /di'fektɪv/
a. 有缺点的, 有缺陷的, 有毛病的

¹ **CAPTCHA**: 全自动区分计算机和人类的图灵测试, 验证码 (Completely Automated Public Turing Test to Tell Computers and Humans Apart 的缩略), 读作 /'kæptʃə/。

² **Turing test**: 图灵测试。英国数学家和逻辑学家艾伦·图灵 (1912—1954) 1950 年提出的一种测试方法, 用来检测一台计算机是否具有类似人类思维的能力。

³ *A tax on new features might help, but that might be a tough sell in some quarters.*: 对新特征征税可能发挥作用, 但这样做可能让有些人士很难接受。

slot / slot /

n. 狭长孔, 狭缝

slot machine

吃角子老虎 (一种投硬币的赌具)

roulette / ru:'let /

n. 轮盘赌

roulette wheel

(轮盘赌台上的) 轮盘

blackjack

/ 'blækdʒæk /

n. 21 点 (一种牌戏)

dealer / 'di:lə /

n. 发牌者; 商人

casino / kə'si:nəu /

n. (有表演、舞池等的) 卡西诺赌场

nice, except it would bankrupt the entire software industry in the first year.

Computer networks raise new legal problems when they interact with old laws. Electronic gambling provides an example. Computers have been simulating things for decades, so why not simulate slot machines, roulette wheels, blackjack dealers, and more gambling equipment? Well, because it is illegal in a lot of places. The trouble is, gambling is legal in a lot of other places (England, for example) and casino owners there have grasped the potential for Internet gambling. What happens if the gambler, the casino, and the server are all in different countries, with conflicting laws? Good question.

Exercises

I. Fill in the blanks with the information given in the text:

1. The argument for communications not to be differentiated by their content or source or who is providing the content is known as network _____.
2. Social _____, message boards, _____ sharing sites, and a host of other applications allow people to share their views with like-minded individuals.
3. CAPTCHAs are used by a computer to ask a person to solve a short _____ task to show that he is human.
4. _____ messages masquerade as originating from a trustworthy party to try to trick you into revealing sensitive information.

II. Translate the following terms or phrases from English into Chinese and vice versa:

- | | |
|----------------------|----------|
| 1. message board | 5. 过滤软件 |
| 2. software vendor | 6. 版权侵犯 |
| 3. anonymous message | 7. 网络中立性 |
| 4. video clip | 8. 网络运营商 |

Unit 12 Electronic Commerce

(电子商务)

Section A

Electronic Commerce



I. Introduction

downturn

/ˈdaunte:n/

n. 衰退; 下降趋势

dot-com /ˈdotkɒm/

n. 网络公司

bust /bʌst/

n. 崩溃, 不景气

dot-bomb /ˈdotbɒm/

n. 网络炸弹

The business phenomenon that we now call electronic commerce has had an interesting history. From humble beginnings in the mid-1990s, electronic commerce grew rapidly until 2000, when a major **downturn** occurred. Many people have seen news stories about the “dot-com boom” followed by the “dot-com **bust**” or the “dot-bomb”. Beginning in 2003, electronic commerce began to show signs of new life. Although the rapid expansion and high levels of investment of the boom years are not likely to

be repeated, the second wave of electronic commerce is well underway.

II. Electronic Commerce and Electronic Business

To many people, the term “electronic commerce” means shopping on the part of the Internet called the World Wide Web (the Web). However, electronic commerce (or e-commerce) also includes many other activities, such as businesses trading with other businesses and internal processes that companies use to support their buying, selling, hiring, planning, and other activities. Some people use the term “electronic business” (or e-business) when they are talking about electronic commerce in this broader sense. Most people use the terms “electronic commerce” and “electronic business” interchangeably. Here the term “electronic commerce” is used in its broadest sense and includes all business activities conducted using electronic data transmission technologies. The most common technologies used are the Internet and the World Wide Web, but other technologies, such as wireless transmissions on **mobile telephone** and **personal digital assistant** (PDA) devices, are also included.

mobile telephone
移动电话
**personal digital
assistant**
个人数字助理

III. Categories of Electronic Commerce

Some people find it useful to categorize electronic commerce by the types of entities participating in the transactions or business processes. The five general electronic commerce categories are business-to-consumer, business-to-business, business processes, consumer-to-consumer, and business-to-government. The three categories that are most commonly used are:

- Consumer shopping on the Web, often called business-to-consumer (or B2C¹)
- Transactions conducted between businesses on the Web, often called business-to-business (or B2B²)
- Transactions and business processes that companies, governments, and other organizations undertake on the Internet to support selling and purchasing activities

Figure 12A-1 shows the three main elements of electronic commerce. The figure presents a rough approximation of the relative sizes of these

¹ B2C: 企业对消费者 (business-to-consumer 的缩略)。在计算机英语中, 一些术语有用发音相同的数字代替单词的现象。电子商务中的 B2C、C2C、B2B 和 B2G 中的 2 均代替 to。再如, 在 XML4J (eXtensible Markup Language for Java, 用于 Java 的可扩展标记语言) 中, 4 代替 for。

² B2B: 企业对企业 (business-to-business 的缩略)。

elements. In terms of dollar volume and number of transactions, B2B electronic commerce is much greater than B2C electronic commerce. However, the number of supporting business processes is greater than the number of all B2C and B2B transactions combined. The large oval in Figure 12A-1 that represents the business processes that support selling and purchasing activities is the largest element of electronic commerce.

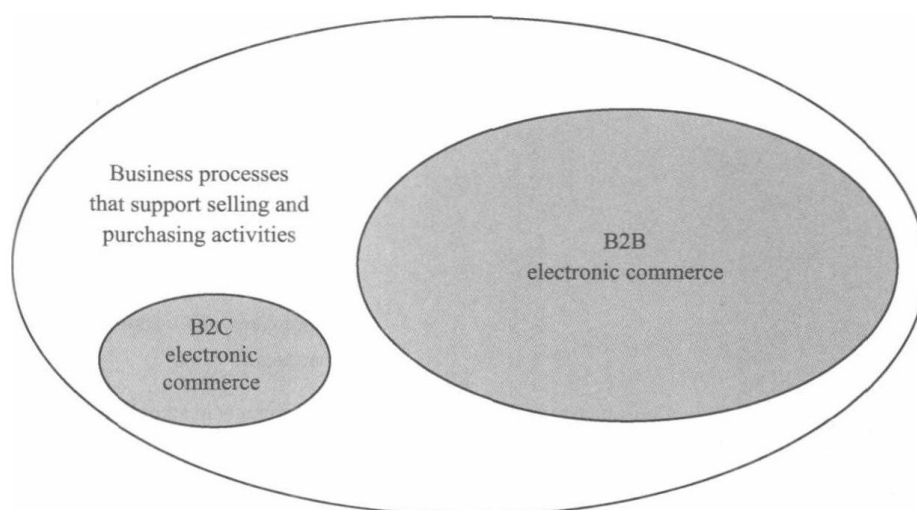


Figure 12A-1: Elements of Electronic Commerce

Some researchers define a fourth category of electronic commerce, called consumer-to-consumer (or C2C¹), which includes individuals who buy and sell items among themselves. For example, C2C electronic commerce occurs when a person sells an item through a Web auction site to another person.

Finally, some researchers also define a category of electronic commerce called business-to-government (or B2G²); this category includes business transactions with government agencies, such as paying taxes and filing required reports³. An increasing number of states have Web sites that help companies do business with state government agencies. For example, the CAL-Buy⁴ site makes it easy for businesses to conduct online transactions with the State of California. Figure 12A-2 summarizes these five categories of electronic commerce.

¹ C2C: 消费者对消费者 (consumer-to-consumer 的缩略)。

² B2G: 企业对政府 (business-to-government 的缩略)。

³ filing required reports: 发送所要求的报告。

⁴ CAL-Buy: 美国最受欢迎的网上政府采购服务网站, 由加利福尼亚州开办。

	Category	Description	Example
	Business-to-consumer (B2C)	Businesses sell products or services to individual consumers.	Walmart.com ¹ sells merchandise to consumers through its Web site.
	Business-to-business (B2B)	Businesses sell products or services to other businesses.	Grainger.com ² sells industrial supplies to large and small businesses through its Web site.
	Business processes that support buying and selling activities	Businesses and other organizations maintain and use information to identify and evaluate customers, suppliers, and employees. Increasingly, businesses share this information in carefully managed ways with their customers, suppliers, employees, and business partners.	Dell Computer uses secure Internet connections to share current sales and sales forecast information with suppliers. The suppliers can use this information to plan their own production and deliver component parts to Dell in the right quantities at the right time.
	Consumer-to-consumer (C2C)	Participants in an online marketplace can buy and sell goods to each other. Because one party is selling, and thus acting as a business, we treat C2C transactions as part of B2C electronic commerce.	Consumers and businesses trade with each other in the eBay.com ³ online marketplace.
	Business-to-government (B2G)	Businesses sell goods or services to governments and government agencies. We treat B2G transactions as part of B2B electronic commerce.	CAL-Buy portal allows businesses to sell online to the State of California.
marketplace /'mɑ:kɪtpleɪs/ n. 市场, 集市, 市集			
portal /'pɔ:təl/ n. 门户, 入口			

Figure 12A-2: Electronic Commerce Categories

IV. The Second Wave of Electronic Commerce

Economists Chris Freeman⁴ and Francisco Louçã⁵ describe four waves that occurred in the Industrial Revolution in their book *As Time Goes By*. Many researchers predict that electronic commerce and the information revolution brought about by the Internet will go through similar waves.

¹ *Walmart.com*: 沃尔玛公司 (Wal-Mart Stores, Inc.) 的电子商务网站。沃尔玛是世界上最大的连锁零售商, 创办于 1962 年, 总部设在美国阿肯色州本顿维尔市 (Bentonville)。

² *Grainger.com*: 固安捷公司 (W. W. Grainger, Inc.) 的电子商务网站。固安捷是世界上著名的工业品分销商, 成立于 1927 年, 总部设在美国伊利诺伊州莱克福里斯特市 (Lake Forest)。

³ *eBay*: 电子港湾, 著名的电子商务网站, 1995 年在美国加利福尼亚州圣何塞市 (San Jose) 创办。

⁴ *Chris Freeman*: 克里斯·弗里曼 (1921—2010), 英国经济学家。

⁵ *Francisco Louçã*: 弗朗西斯科·卢桑 (1956—), 葡萄牙政治家与经济学家。

Those researchers agree that the second wave of electronic commerce has begun.

The first wave of electronic commerce was primarily a U.S. phenomenon. Web pages were primarily in English, particularly on commerce sites. As the second wave begins, it is clear that the future of electronic commerce will be international in scope and will allow sellers to do business in many countries and in many languages. The problems of language translation and handling currency **conversion** will need to be solved to allow efficient conduct of business in the second wave.

conversion

/kən'veɪʃən/

n. 转换; 转变;
兑换

In the first wave, easy access to start-up capital led to an over-emphasis on creating new large enterprises to exploit electronic commerce opportunities.¹ Investors were excited about electronic commerce and wanted to participate, no matter how much it cost or how bad the underlying ideas were. In the second wave, established companies are using their own internal funds to finance gradual expansion of electronic commerce opportunities. These **measured** and carefully considered investments are helping electronic commerce grow more steadily, although more slowly.

measured /'meɪʒəd/

a. 慢而稳的; 慎重的, 几经斟酌的

The Internet technologies used in the first wave, especially in B2C commerce, were slow and inexpensive. Most consumers connected to the Internet using dial-up modems. The increase in broadband connections in homes is a key element in the second wave. Although these connections are more expensive, they are more than 10 times faster. This increased speed not only makes Internet use more efficient, it can alter the way people use the Web.

bar code

条形码

assembly /ə'sembli/

n. 部件; 组合体

biometric(al)

/ˌbaɪəʊ'metrik(əl)/

a. 生物统计学的

fingerprint

/'fɪŋɡəprɪnt/

n. 指纹(印), 手印

In the first wave, Internet technologies were integrated into B2B transactions and internal business processes by using **bar codes** and scanners to track parts, **assemblies**, inventories, and production status. These tracking technologies were not well integrated. Also, companies sent transaction information to each other using a patchwork of communication methods including fax, e-mail, and electronic data interchange (EDI²). In the second wave, radio-frequency identification (RFID) devices and smart cards are being combined with **biometric** technologies such as **fingerprint**

¹ *In the first wave, easy access to start-up capital led to an overemphasis on creating new large enterprises to exploit electronic commerce opportunities.*: 在第一次浪潮中, 创办资金比较容易得到, 这导致人们过分强调创办新的大型企业, 以利用电子商务机会。

² *EDI*: 电子数据交换 (electronic data interchange 的首字母缩略)。

fingerprint reader

指纹读取器

retina /'retinə/

n. 视网膜

retina scanner

视网膜扫描仪

readers and retina scanners to control more items and people in a wider variety of situations. These technologies are increasingly integrated with each other and with communication systems that allow companies to communicate with each other and share transaction, inventory level, and customer demand information effectively.

unstructured

/ˌʌn'strʌktʃəd/

a. 非结构化的, 没有特定结构(或系统)的, 松散的

dip /dɪp/

n. (暂时或小幅度的)减少; 下降

The use of electronic mail (or e-mail) in the first wave was as a tool for relatively **unstructured** communication. In the second wave, sellers are using e-mail as an integral part of their marketing and customer contact strategies.

Online advertising was the main revenue source of many failed dot-com businesses in the first wave. After a two-year **dip** in online advertising activity and revenues, companies are beginning the second wave with a renewed interest in making the Internet work as an effective advertising medium. Some categories of online advertising, such as employment services, are growing rapidly and are replacing traditional advertising outlets.

fraught /frɔ:t/a. [一般作表语] 充满的 (*with*)

The sale of digital products was **fraught** with difficulties during the first wave of electronic commerce. The music recording industry was unable (or, some would say, unwilling) to devise a way to distribute digital music on the Web. This created an environment in which digital piracy—the theft of musical artists' **intellectual property**—became **rampant**. The promise of electronic books was also unfulfilled. The second wave offers the promise of legal distribution of music, video, and other digital products on the Web. Apple Computer's iTunes¹ site is one of the early second wave attempts at digital product distribution.

intellectual property

知识产权

rampant

/'ræmpənt/

a. 猖獗的; 无约束的

Figure 12A-3 shows a summary of some key characteristics of the first wave and the second wave of electronic commerce. This list is not complete because every day brings new technologies and combinations of existing technologies that make additional second wave opportunities possible.

Not all of the future of electronic commerce is based in its second wave. Some of the first wave companies were successful, such as Amazon.com², eBay, and Yahoo!. The second wave of electronic commerce

¹ **iTunes**: 苹果公司开发的一种数字媒体播放器应用程序, 有自己的网上商店, 供用户购买和下载数字音乐、视频图像等。

² **Amazon.com**: 亚马逊网站, 由亚马逊公司 (Amazon.com, Inc.) 开办的电子商务网站, 最初是作为网上书店, 不久就使其在线销售产品多元化。亚马逊公司成立于 1994 年, 1995 年开始在线经营, 总部设在美国华盛顿州西雅图市。

will provide new opportunities for these businesses, too.

	Electronic Commerce Characteristic	First Wave	Second Wave
disparate /'dispərit / a. 完全（或根本） 不同的；不相干的 customized / 'kʌstəmaɪzd / a. 定制的， 用户化的	International character of electronic commerce	Dominated by U.S. companies	Global enterprises in many countries participating in electronic commerce
	Languages	Most electronic commerce Web sites in English	Many electronic commerce Web sites available in multiple languages
	Funding	Many new companies started with outside investor money	Established companies funding electronic commerce initiatives with their own capital
	Connection technologies	Many electronic commerce participants using slow Internet connections	Rapidly increasing use of broadband technologies for Internet connections
	B2B technologies	B2B electronic commerce relied on a patchwork of disparate communication and inventory management technologies	B2B electronic commerce increasingly integrated with radio-frequency identification and biometric devices to manage information and product flows effectively
	E-mail contact with customers	Unstructured e-mail communication with customers	Customized e-mail strategies now integral to customer contact
	Advertising and electronic commerce integration	Over-reliance on simple forms of online advertising as main revenue source	Use of multiple sophisticated advertising approaches and better integration of electronic commerce with existing business processes and strategies
	Distribution of digital products	Widespread piracy due to ineffective distribution of digital products	New approaches to the sale and distribution of digital products

Figure 12A-3: Key Characteristics of the First Two Waves of Electronic Commerce

Exercises

I. Fill in the blanks with the information given in the text:

1. The year 2000 was a turning point in electronic commerce. It marked the end of the “dot-com _____.”

2. Most people use the terms “electronic commerce” and “electronic _____” interchangeably.

3. Based on the types of participating _____, electronic commerce can be classified into five categories: B2C, B2B, business processes, C2C, and _____.

4. The three most commonly used categories of electronic commerce are B2C, _____, and business processes.

5. In C2C transactions, one party is selling, and thus acting as a business, so they are sometimes treated as part of _____ electronic commerce.

6. The first wave of electronic commerce was dominated by _____ companies.

7. In the first wave, there was an overemphasis on creating new large enterprises to exploit electronic commerce opportunities because of the easy access to start-up _____.

8. The increase in _____ connections in homes is a key element in the second wave of electronic commerce.

II. Translate the following terms or phrases from English into Chinese and vice versa:

- | | |
|----------------------------------|---------------|
| 1. customized marketing strategy | 11. 射频识别装置 |
| 2. B2G transaction | 12. 电子数据交换 |
| 3. dial-up modem | 13. 库存管理技术 |
| 4. dot-com bust | 14. 知识产权 |
| 5. smart card | 15. 条形码 |
| 6. digital piracy | 16. 货币兑换 |
| 7. dot-com boom | 17. 电子图书 |
| 8. C2C transaction | 18. 视网膜扫描仪 |
| 9. Web auction site | 19. 个人数字助理 |
| 10. fingerprint reader | 20. 企业对企业电子商务 |

III. Fill in each of the blanks with one of the words given in the following list, making changes if necessary:

- | | | | |
|------------------|--------------------|------------------|----------------|
| <i>effective</i> | <i>use</i> | <i>business</i> | <i>adopt</i> |
| <i>legal</i> | <i>application</i> | <i>challenge</i> | <i>product</i> |

<i>marketing</i>	<i>human</i>	<i>technology</i>	<i>nature</i>
<i>more</i>	<i>integrate</i>	<i>identify</i>	<i>end</i>

Commerce, the negotiated exchange of goods or services, has been practiced in traditional ways for thousands of years. Electronic commerce is the _____ of new technologies, particularly Internet and Web _____, to help individuals, businesses, and other organizations conduct business more _____. As in the Industrial Revolution, electronic commerce will be _____ in waves of change. The first wave of electronic commerce _____ in 2000. More recently, a second wave with new approaches to _____ Internet technologies into business processes has begun. In the second wave, _____ are focusing less on overall business models and _____ on improving specific business processes.

Using electronic commerce, some businesses have been able to create new _____ and services, and others have improved the promotion, _____, and delivery of existing *offerings* (待售品). Firms have also found many ways to _____ electronic commerce to improve purchasing and supply activities; _____ new customers; and operate their finance, administration, and _____ resource management activities more efficiently. The inherently global _____ of electronic commerce leads to many opportunities and a few _____. Businesses must be careful to understand the trust, cultural, language, and _____ issues that arise in international business.

IV. Translate the following passage from English into Chinese:

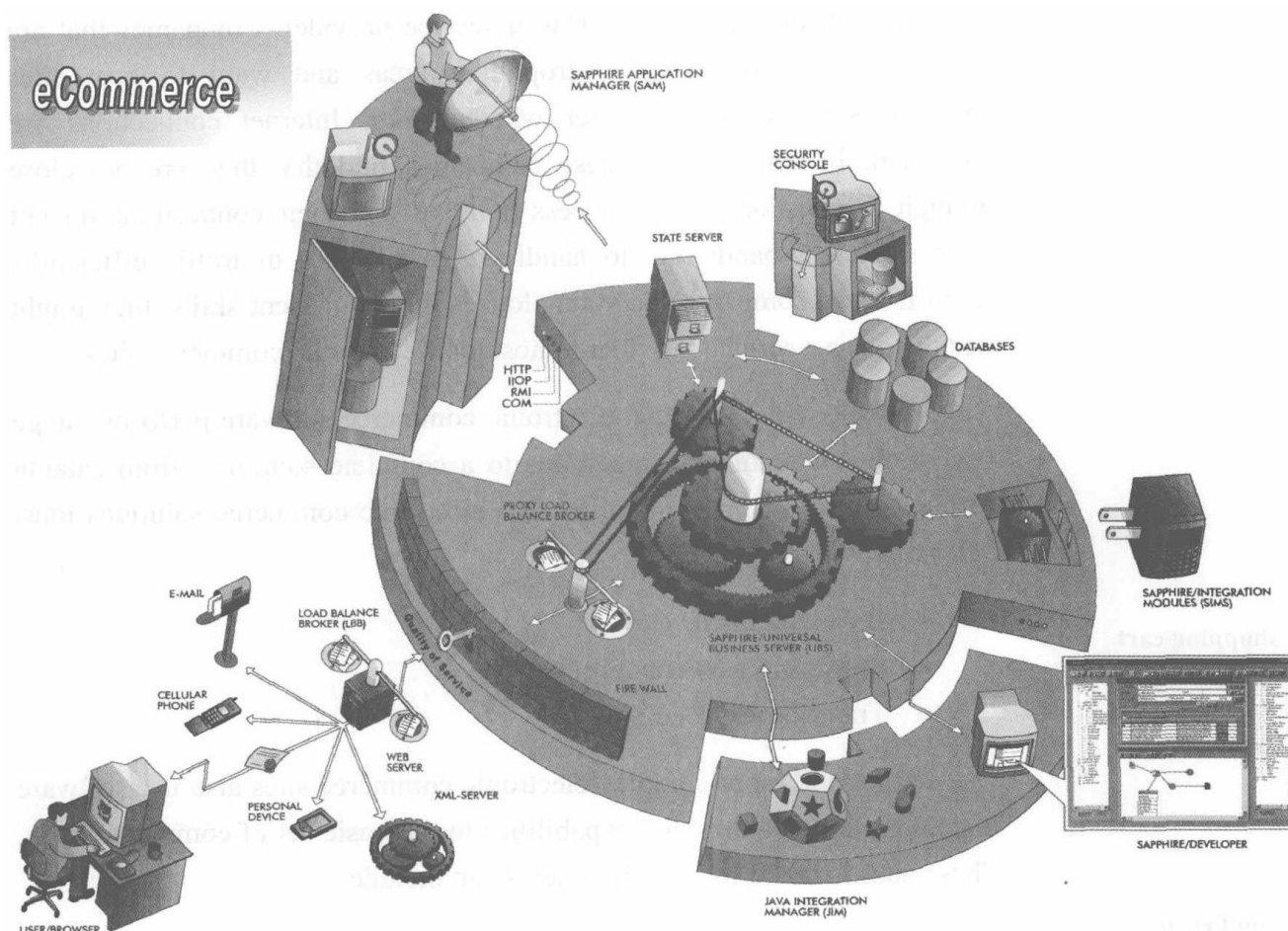
Firms are interested in electronic commerce because, quite simply, it can help increase profits. All the advantages of electronic commerce for businesses can be summarized in one statement: Electronic commerce can increase sales and decrease costs. Advertising done well on the Web can get even a small firm's promotional message out to potential customers in every country in the world. A firm can use electronic commerce to reach small groups of customers that are geographically scattered. The Web is particularly useful in creating virtual communities that become ideal target markets for specific types of products or services.

Just as electronic commerce increases sales opportunities for the seller, it increases purchasing opportunities for the buyer. Businesses can use electronic commerce to identify new suppliers and business partners. Negotiating price and delivery terms is easier in electronic commerce

because the Internet can help companies efficiently obtain competitive bid information. Electronic commerce increases the speed and accuracy with which businesses can exchange information, which reduces costs on both sides of transactions.

Section B

Basic Functions of Electronic Commerce Software



I. Introduction

The size and objectives of electronic commerce sites vary greatly; thus, a variety of software and hardware products are used to build those sites. At the inexpensive end of the spectrum of electronic commerce solutions are choices such as externally hosted stores that provide software tools to build an online store on a host's site.¹ At the other end of the range are sophisticated electronic commerce **software suites** that can handle high transaction volumes and include a broad **assortment** of features and tools.

software suite

软件套件

assortment

/ ə'sɔ:tment /

n. 分类; 各种各样

¹ *At the inexpensive end of the spectrum of electronic commerce solutions are choices such as externally hosted stores that provide software tools to build an online store on a host's site.* 在电子商务解决方案谱系的低成本端是一些诸如外部托管商店的选择, 这种外部托管商店提供软件工具, 用于在主机站点搭建一个在线商店。

in-house /'in'haus/
ad. 在机构内部; 无
外援地
a. 内部的, 自用的
hosting /'həʊstɪŋ/
n. 托管

An early decision is whether the company should use an external host or host the electronic commerce site **in-house**. Companies that have an existing information technology (IT) staff of programmers, Web designers, and network engineers are more likely to choose an in-house **hosting** approach. If a company does not have or cannot easily hire people with the skills required to set up and maintain an electronic commerce site, it can outsource all or part of the job to a service provider. Companies that are located outside of major metropolitan areas and want to host sites themselves must also consider whether their Internet connections are sufficient. In many cases, these companies find that they are not close enough to a major Internet access point or that their connections do not have sufficient bandwidth to handle large volumes of traffic efficiently. Even if these companies have employees with sufficient skills, they might decide to use a service provider to host their electronic commerce sites.

The specific duties that electronic commerce software performs range from a few fundamental operations to a complete solution—from catalog display to fulfillment notification. All electronic commerce solutions must at least provide:

shopping cart
购物车

- A catalog display
- Shopping cart capabilities
- Transaction processing

Larger and more complex electronic commerce sites also use software that adds other features and capabilities to the basic set of commerce tools. These additional software components can include:

middleware
/'mi:dlweə/
n. 中(间)件,
中间设备
accounting
/ə'kaʊntɪŋ/
n. 会计(制度); 记
账; 结账; 结算

- Middleware that integrates the electronic commerce system with existing company information systems that handle inventory control, order processing, and accounting
- Enterprise application integration
- Web services
- Integration with enterprise resource planning (ERP) software
- Supply chain management (SCM) software
- Customer relationship management (CRM) software
- Content management software
- Knowledge management software

Tools required by all electronic commerce sites are described in the following sections.

II. Catalog Display

offering

/ˈɔfərɪŋ; ˈɔ:-/

n. 待售品; 提供之物; 捐献物

footwear /ˈfʊtwɛə/

n. [总称] 鞋类

athletic /æθˈletɪk/

a. 运动的, 体育的

listing /ˈlɪstɪŋ/

n. 列表; 一览表;

目录

A catalog organizes the goods and services being sold. To further organize its offerings, a retailer may break them down into departments. As in a physical store, merchandise in an online store can be grouped within logical departments to make locating an item, such as a camping stove, simpler. Web stores often use the same department names as their physical counterparts.¹ In most physical stores, each product is kept in only one place. A Web store has the advantage of being able to include a single product in multiple categories. For example, running shoes can be listed as both **footwear** and **athletic gear**.

A small commerce site can have a very simple static catalog. A catalog is a **listing** of goods and services. A static catalog is a simple list written in HTML that appears on a Web page or a series of Web pages. To add an item, delete an item, or change an item's listing, the company must edit the HTML of one or more pages. Larger commerce sites are more likely to use a dynamic catalog. A dynamic catalog stores the information about items in a database, usually on a separate computer that is accessible to the server that is running the Web site itself. A dynamic catalog can feature multiple photos of each item, detailed descriptions, and a search tool that allows customers to search for an item and determine its availability. The software that implements a dynamic catalog is often included in larger electronic commerce software packages; however, some companies write their own software to link their existing databases of product information to their Web sites.

Small Web stores that sell fewer than 30 or 40 items need only a simple list of products or categories. Organization of the items is not particularly important. Companies that offer only a small number of items can provide a photo of each item on the Web page that is a link to more information about the product. A static catalog is sufficient for their needs. Larger electronic commerce sites require the more sophisticated navigation aids and better product organization tools that are a part of dynamic catalogs.

Good sites give buyers alternative ways to find products. Besides offering a well-organized catalog, large sites with many products can

¹ *Web stores often use the same department names as their physical counterparts.*: 网络商店常常使用与实际商店相同的门类名称。

provide a search engine that allows customers to enter descriptive search terms, such as “men’s shirts,” so they can quickly find the Web page containing what they want to purchase. Remember, the most important rule of all commerce is: Never stand in the way of a customer who wants to buy something.

III. Shopping Cart

text box

文本框

list box

列表框

stock-keeping unit

库存单位

In the early days of electronic commerce, shoppers selected items they wanted to purchase by filling out online forms. Using **text box** and **list box** from controls to indicate their choices, users entered the quantity of an item in the quantity text box, the SKU (**stock-keeping unit**) or product number in another text box, and the unit price in yet another text box.¹ This system was awkward for ordering more than one or two items at a time.

One problem with forms-based shopping was that shoppers had to write down product codes, unit prices, and other information about the product before going to the order form, which was inevitably on another page. Another problem was that customers sometimes forgot whether they had clicked the submit button to send in their orders. As a result, they either sent the same order twice (pressing the submit button when they had already done so) or thought they had submitted the order when they really had not (consequently failing to submit the order). The forms-based method of shopping was confusing and error **prone**.

prone / prəʊn /

a. 有…倾向的,
易于…的

The forms-based method of ordering has given way to electronic shopping carts. Today, shopping carts are a standard of electronic commerce. A shopping cart, also sometimes called a shopping bag or shopping basket, keeps track of the items the customer has selected and allows customers to view the contents of their carts, add new items, or remove items. To order an item, the customer simply clicks that item. All of the details about the item, including its price, product number, and other identifying information, are stored automatically in the cart. If a customer later changes his or her mind about an item, he or she can view the cart’s contents and remove the unwanted items. When the customer is ready to conclude the shopping session,² the **click** of a button executes the purchase transaction. Figure

click / klik /

n. (鼠标的) 点击

¹ Using text box and list box from controls to indicate their choices, users entered the quantity of an item in the quantity text box, the SKU (stock-keeping unit) or product number in another text box, and the unit price in yet another text box.: 用户使用控件上的文本框和列表框来表明他们的选择, 在数量文本框中输入一种商品的数量, 在另一个文本框中输入库存单位或产品编号, 并在第三个文本框中输入单价。SKU 或读作 / skju: / 或按各字母的读音读。

² to conclude the shopping session: 结束这次购物。

12B-1 shows a typical shopping basket page at an electronic commerce site.

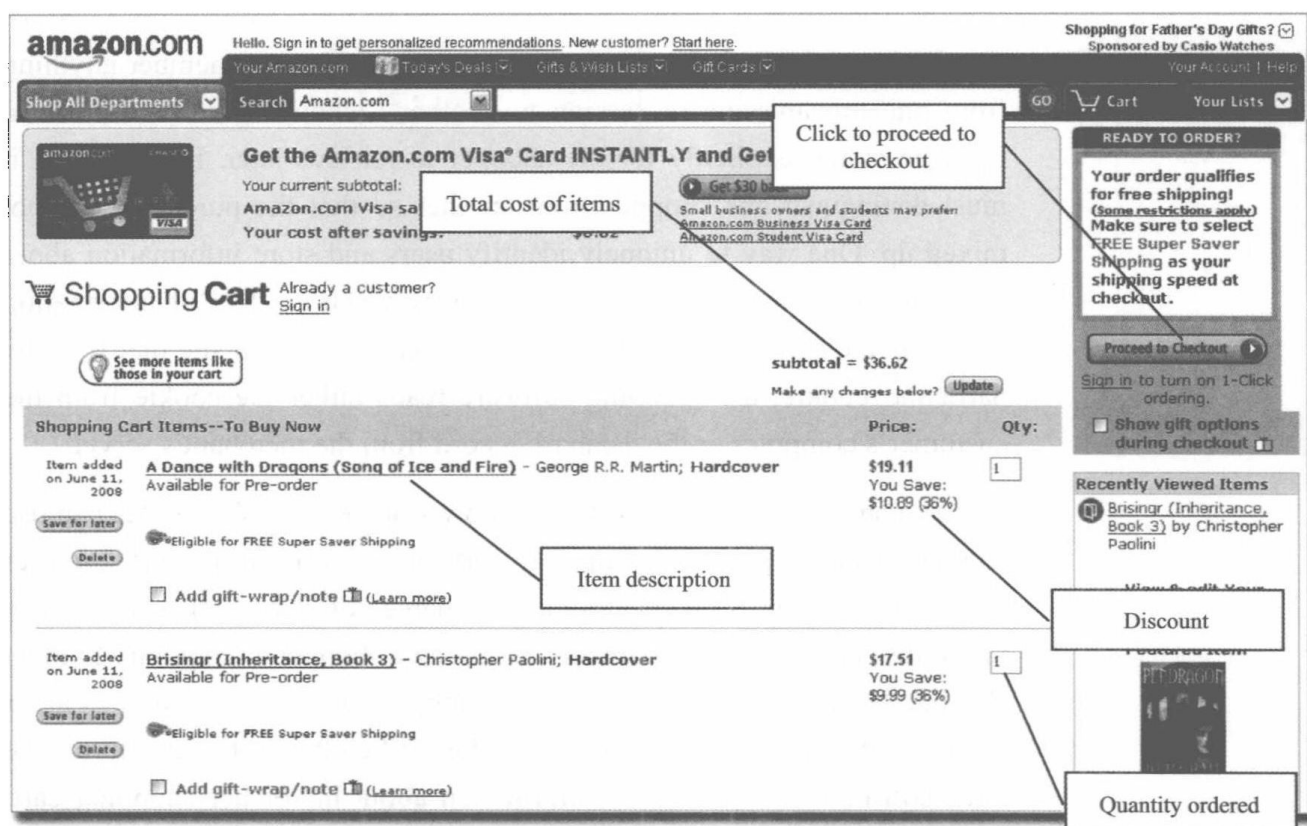


Figure 12B-1: A Typical Shopping Cart Page

Clicking the checkout button usually displays a screen that asks for billing and shipping information and that confirms the order. The shopping cart software keeps a running total of each type of item.¹ The shopping cart calculates a total as well as sales tax and shipping costs.

Shopping cart software at some Web commerce sites allows the customer to fill a shopping cart with purchases, put the cart in virtual storage, and come back days later to confirm and pay for the purchases. A number of companies, including Biznet Internet Services², CartIt!³, SalesCart⁴, and WebGenie Software⁵, sell shopping cart software that sellers

¹ The shopping cart software keeps a running total of each type of item.: 购物车软件持续累计出每种商品的总和。

² Biznet Internet Services: Biznet 互联网服务公司 (Biznet Internet Services, Inc.), 一家以提供网站建设服务、数据服务和主机服务而闻名的公司, 成立于 1995 年, 总部设在美国弗吉尼亚州里士满市。

³ CartIt!: 一家提供电子商务相关服务的公司。

⁴ SalesCart: ComCity 公司开发的一种电子商务站点设计工具。ComCity 成立于 1995 年, 是一家 Internet presence provider (首字母缩写为 IPP, 汉译为“因特网平台提供商”、“互联网平台提供商”等), 总部设在美国得克萨斯州奥斯丁市。

⁵ WebGenie Software: WebGenie 软件公司 (WebGenie Software Pty Ltd.), 一家提供网上购物车软件和电子商务相关支持软件的澳大利亚公司。

can add to their Web sites. These software packages range in price from a few hundred dollars to several thousand dollars, plus an ongoing monthly fee.¹

stateless system
无状态系统

Because the Web is a **stateless system**—unable to remember anything from one transmission or session to another²—shopping cart information must be stored explicitly for the shopper to retrieve later. Furthermore, it must distinguish one shopper from another so that the purchases are not mixed up. One way to uniquely identify users and store information about their choices is to create and store cookies, which are bits of information stored on a client computer. When a customer returns to a site that issued a particular cookie, the shopping software reads either the cookie from the customer's computer or the database record from the merchant's server.

If a shopper's browser does not allow storage of cookies, sites can use another way to preserve shopping cart information from one browser session to another. Some electronic commerce software packages, such as ShopSite³, do this by automatically assigning a shopper a temporary number. The number is added to the end of the shopper's URL and persists as he or she navigates from one Web site to another. When the customer returns, the URL still contains the bits of information about his or her shopping cart. When the customer closes the browser, the temporary number is discarded and thus cannot be reused, even if the customer later reopens the browser and returns to the same Web site.

IV. Transaction Processing

Transaction processing occurs when the shopper proceeds to the virtual checkout counter by clicking a checkout button.⁴ Then the electronic commerce software performs any necessary calculations, such as volume discounts⁵, sales tax, and shipping costs. At checkout, the customer's Web browser software and the seller's Web server software both switch into a secure state of communication.

Transaction processing can be the most complex part of the online sale. Computing taxes and shipping costs are important parts of this process, and

¹ *plus an ongoing monthly fee*: 加上不断支付的月费。

² *the Web is a stateless system—unable to remember anything from one transmission or session to another*: 万维网是一种无状态的系统, 从一个(信息)传送或会话转到另一个时, 无法记住任何事(即原有的状态无法保存)。

³ *ShopSite*: 一种供中小型企业使用的电子商务购物车软件, 由 ShopSite 公司(ShopSite, Inc.)出品。ShopSite 公司设在美国犹他州奥勒姆市(Orem), 是中小型企业电子商务购物车及目录软件提供商。

⁴ *Transaction processing occurs when the shopper proceeds to the virtual checkout counter by clicking a checkout button.*: 购物者通过点击结账按钮前往虚拟付款台时, 交易处理过程便开始了。

⁵ *volume discounts*: 大数量折扣, 即因买的数量多而给予的折扣。

site administrators must continually check tax rates and shipping tables to make sure they are current.¹ Some software enables the Web server to obtain updated shipping rates by connecting directly to shipping companies to retrieve information.

complication

/ ,kɒmpli'keɪʃən /

n. 复杂情况; 困难, 难题

coupon / 'ku:pən /

n. 购物优惠券, 礼券, 赠券

Other calculation **complications** include provisions for **coupons**, special promotions, and time-sensitive offers;² for example, “purchase a round-trip ticket before the end of the month and receive a 50 percent discount.” Some shopping cart software designed for small and midsize companies provides connections to accounting software so that Web sales can be entered simultaneously in the company’s accounting system. In larger companies, the integration of the Web site’s transaction processing into the accounting and operation-control systems of the company can be very complex.

Exercises

I. Fill in the blanks with the information given in the text:

1. All electronic commerce sites must at least provide a catalog display, shopping cart capabilities, and _____ processing.
2. Small commerce sites can use a(n) _____ catalog while larger commerce sites are more likely to use a(n) _____ catalog.
3. A shopping cart is sometimes called a shopping _____ or shopping _____.
4. At checkout, the online shopper’s Web browser software and the seller’s Web server software both switch into a _____ state of communication.

II. Translate the following terms or phrases from English into Chinese and vice versa:

- | | |
|-----------------------------|-------------|
| 1. software suite | 6. 动态目录 |
| 2. text box | 7. 购物车软件 |
| 3. virtual checkout counter | 8. 供应链 |
| 4. static catalog | 9. 企业资源计划软件 |
| 5. browser session | 10. 税率 |

¹ *check tax rates and shipping tables to make sure they are current:* 核对税率和运费表, 以确保它们是现行税率和运费表。

² *Other calculation complications include provisions for coupons, special promotions, and time-sensitive offers:* 其他的计算困难包括有关购物优惠券的规定、特别促销和有时限的报价。

Section C

Online Shopping



I. Introduction

evoke /i'vəuk/
v. 唤起; 使人想起
mortar /'mɔ:tə/
n. 砂浆, 灰浆

metaphor /'metəfe/
n. 隐喻; 比喻
storefront
/'stɔ:frʌnt/
n. 店面, 铺面

Online shopping is the process consumers go through to purchase products or services over the Internet. An online shop, eshop, e-store, internet shop, webshop, webstore, online store, or virtual store **evokes** the physical analogy of buying products or services at a bricks-and-mortar retailer or in a shopping mall.

The **metaphor** of an online catalog is also used, by analogy with mail order catalogs.¹ All types of stores have retail websites, including those that do and do not also have physical **storefronts** and paper catalogs.

Online shopping is a type of electronic commerce used for business-to-business (B2B) and business-to-consumer (B2C) transactions.

¹ *The metaphor of an online catalog is also used, by analogy with mail order catalogs.*: 此外, 还通过与邮购商品目录的类比使用在线商品目录的比喻。

II. Logistics

Consumers find a product of interest by visiting the website of the retailer directly, or do a search across many different vendors using a shopping search engine.

Once a particular product has been found on the website of the seller, most online retailers use shopping cart software to allow the consumer to accumulate multiple items and to adjust quantities, by analogy with filling a physical shopping cart or basket in a conventional store. A “checkout” process follows (continuing the physical-store analogy) in which payment and delivery information is collected, if necessary. Some stores allow consumers to sign up for a permanent online account so that some or all of this information only needs to be entered once. The consumer often receives an e-mail confirmation once the transaction is complete. Less sophisticated stores may rely on consumers to phone or e-mail their orders (though credit card numbers are not accepted by e-mail, for security reasons).

1. Payment

Online shoppers commonly use credit cards to make payments; however, some systems enable users to create accounts and pay by alternative means, such as:

cash on delivery

货到付款,
交货付款

wire transfer

电汇

delivery on

payment

付款交货

postal money order

邮政汇票

gift card

礼物卡, 折扣卡

- Debit card
- Various types of electronic money
- **Cash on delivery** (COD, offered by very few online stores)
- Check
- **Wire transfer/delivery on payment**
- **Postal money order**
- PayPal¹
- Google Checkout²
- Moneybookers³
- Reverse SMS⁴ billing to mobile phones⁵
- **Gift cards**

¹ **PayPal**: PayPal 公司 (PayPal, Inc.) 提供的在线收付款工具。PayPal 公司成立于 1998 年, 总部设在美国加利福尼亚州圣何塞市, 2002 年被 eBay 收购。

² **Google Checkout**: Google 公司提供的在线支付处理服务, 意在简化在线购买的支付过程。

³ **Moneybookers**: Moneybookers 公司 (Moneybookers Ltd.) 提供的在线收付款工具。Moneybookers 公司成立于 2001 年, 总部设在英国伦敦。

⁴ **SMS**: 短信服务 (Short Message Service 的首字母缩略)。

⁵ **Reverse SMS billing to mobile phones**: 手机短信服务逆向收费, 指由接收短信的手机而非发送短信的手机支付费用。

direct debit

直接借记,
直接付款

- **Direct debit** in some countries

Some sites will not allow international credit cards and billing address and shipping address have to be in the same country in which the sites do their business. Other sites allow customers from anywhere to send gifts anywhere. The financial part of a transaction might be processed in real time (for example, letting the consumer know their credit card was declined before they log off¹), or might be done later as part of the fulfillment process.

2. Product Delivery

Once a payment has been accepted, the goods or services can be delivered in the following ways.

- **Downloading:** This is the method often used for digital media products such as software, music, movies, or images.
- **Shipping:** The product is shipped to the customer's address.
- **Drop shipping:** The order is passed to the manufacturer or third-party **distributor**, who ships the item directly to the consumer, **bypassing** the retailer's physical location to save time, money, and space.
- **In-store pickup:** The customer orders online, finds a local store using **locator software**, and picks up the product at the closest store. This is the method often used in the bricks and clicks business model.²

drop shipping

直达货运

distributor

/di'stribjutə/

n. 经销商; 批发商

bypass /'baipɑ:s/

v. 绕过; 越过

locator software

定位软件

III. Aggregation

High-volume websites, such as Yahoo!, Amazon.com, and eBay, offer hosting services for online stores to small retailers. These stores are presented within an integrated navigation framework. Collections of online stores are sometimes known as virtual shopping malls or online marketplaces. These virtual shopping malls are sometimes created in order to give the consumer the choice of a variety of online **resellers** without actually selling product directly themselves. These sites receive commission

reseller /,ri:'selə/

n. 转售商, 代销商

¹ *log off*: 注销, 退出系统。

² *the bricks and clicks business model*: 砖块加点击商业模式, 砖块加鼠标商业模式。指将传统商店 (bricks) 与因特网交易 (clicks) 结合起来的商业模式, 或者说将线下 (offline) 或传统商业模式与在线 (online) 或因特网商业模式结合起来的商业模式。“bricks and clicks”也称为“clicks and mortar”、“clicks and bricks”等。

affiliation

/əˌfili'eɪʃən/

n. 联系; 从属关系

payments via **affiliation** with the retailers.¹ This gives the retailers another way into the marketplace other than their own websites. Sometimes the metaphor is taken more literally, and a visual representation similar to a conventional shopping mall or village is presented.²

IV. Product Suitability**affiliate** /ə'filiit/

n. 附属机构;

分公司

affiliate marketing

联属网络营销,

联盟营销

Many successful purely virtual companies deal with digital products (including information storage, retrieval, and modification), music, movies, office supplies, education, communication, software, photography, and financial transactions. Examples of this type of company include Google, eBay, and PayPal. Other successful marketers use drop shipping or **affiliate marketing** techniques to facilitate transactions of tangible goods without maintaining real inventory. Examples include numerous sellers on eBay.

shut-in /'ʃʌtɪn/

n. 不能外出的人;

卧病在家的人

Some non-digital products have been more successful than others for online stores. Profitable items often have a high value-to-weight ratio, they may involve embarrassing purchases, they may typically go to people in remote locations, and they may have **shut-ins** as their typical purchasers. Items which can fit through a standard letterbox³—such as music CDs⁴, DVDs, and books—are particularly suitable for a virtual marketer, and indeed Amazon.com, one of the few enduring dot-com companies⁵, has historically concentrated on this field.

centrifugal

/sen'trifjʊəl/

a. 离心的

centrifugal pump

离心泵

outlet /'aʊtlet, -lɪt/

n. 销路; 商店

Products such as spare parts, both for consumer items like washing machines and for industrial equipment like **centrifugal pumps**, also seem good candidates for selling online. Retailers often need to order spare parts specially, since they typically do not stock them at consumer **outlets**—in such cases, e-commerce solutions in spares do not compete with retail stores, only with other ordering systems. A factor for success in this niche can consist of providing customers with exact, reliable information about which part number their particular version of a product needs, for example by providing parts lists keyed by serial number.⁶

¹ *These sites receive commission payments via affiliation with the retailers.*: 这些网站凭借与零售商的联系收取佣金。

² *Sometimes the metaphor is taken more literally, and a visual representation similar to a conventional shopping mall or village is presented.*: 有时, 该比喻更多地按字面理解, 在网站上给出的是类似于传统购物中心或村落的直观表现。

³ *Items which can fit through a standard letterbox*: 可以塞进标准邮政信箱的物品。

⁴ *CD*: 光盘 (compact disc 的首字母缩略)。

⁵ *dot-com companies*: 在网上经营的公司, 这些公司的因特网地址通常带有.com 后缀。

⁶ *which part number their particular version of a product needs, for example by providing parts lists keyed by serial number*: 他们特定版本的产品所需的零部件编号, 例如提供以序列号为索引键值的零部件列表。

fitting /'fitɪŋ/

n. 试穿, 试衣

Products less suitable for e-commerce include products that have a low value-to-weight ratio, products that have a smell, taste, or touch component, products that need trial fittings—most notably clothing—and products where colour integrity appears important.¹

V. Fraud and Security Concerns

Given the lack of ability to inspect merchandise before purchase, consumers are at higher risk of fraud on the part of the merchant than in a physical store. Merchants also risk fraudulent purchases using stolen credit cards or fraudulent repudiation of the online purchase.

fraudulent

/'frɔ:dʒələnt/

a. 欺骗性的, 欺诈性的

repudiation

/riˌpju:di'eɪʃən/

n. 拒绝; 否认

socket /'sɒkɪt/

n. 插座; 插槽; 套接字

Secure Sockets**Layer**

安全套接层(协议)

intercept /,ɪntə'sept/

v. 拦截; 截取

transit

/'trænsɪt, -zɪt/

n. 运输, 运送

identity theft

身份(信息)盗取

deploy /dɪ'plɔɪ/

v. 部署

reputable

/'repjuteɪbəl/

a. 声誉好的

quality seal

质量图章

foil /fɔɪl/

v. 挫败; 使受挫折

Secure Sockets Layer (SSL) encryption has generally solved the problem of credit card numbers being intercepted in transit² between the consumer and the merchant. Identity theft is still a concern for consumers when hackers break into a merchant's website and steal names, addresses, and credit card numbers. Computer security has thus become a major concern for merchants and e-commerce service providers, who deploy countermeasures such as firewalls and antivirus software to protect their networks.

Phishing is another danger, where consumers are fooled into thinking they are dealing with a reputable retailer, when they have actually been manipulated into feeding private information to a system operated by a malicious party.

Quality seals can be placed on an online shop's webpage if it has undergone an independent assessment and meets all the requirements of the company issuing the seal. The purpose of these seals is to increase the confidence of the online shoppers; the existence of many different seals, or seals unfamiliar to consumers, may foil this effort to a certain extent.

Measures that consumers can take to protect themselves when using online retailer services include:

- Sticking with known stores, or attempting to find independent consumer reviews of their experiences; also ensuring that there is comprehensive contact information on the website before using the

¹ products that have a low value-to-weight ratio, products that have a smell, taste, or touch component, products that need trial fittings—most notably clothing—and products where colour integrity appears important: 具有低价值与重量比的产品, 具有与嗅觉、味觉或触觉相关成分的产品, 需要试穿的产品——最显著的是衣着, 以及色彩完整性显得很重要的产品。

² in transit: 在运输中, 在运送中, 在途中。在本句中指“在传输的过程中”。

oversight

/'əʊvəsait /

n. 监督; 照管

trustmark

/'trʌstmɑ:k /

n. 信任标记

privacy policy

隐私政策

service, and noting if the retailer has enrolled in industry oversight programs such as **trustmark** or **trust seal**.

- Ensuring that the retailer has an acceptable **privacy policy** posted. For example, note if the retailer does not explicitly state that it will not share private information with others without consent.
- Ensuring that the vendor's address is protected with SSL when credit card information is entered. If it is, the address on the credit card information entry screen will start with "HTTPS¹".
- Using strong passwords without personal information. Another option is a "pass phrase," which might be something along the lines: "I shop 4 good a buy!!"² It is difficult to hack, and provides a variety of uppercase, lowercase, and special characters and could be site specific and easy to remember.³

Exercises

I. Fill in the blanks with the information given in the text:

1. Online shoppers commonly use _____ cards to make payments.
2. In online shopping, purchased goods or services can be delivered through _____, shipping, drop shipping, and _____ pickup.
3. High-volume websites sometimes create virtual shopping _____ or online marketplaces.
4. Identity theft happens when _____ break into a merchant's website and steal consumers' names, addresses, and credit card numbers.

¹ **HTTPS**: (使用)安全(套接层的)超(级)文本传送(或传输)协议(*Hypertext Transfer Protocol over Secure Sockets Layer* 的缩略)。

² *Another option is a "pass phrase," which might be something along the lines: "I shop 4 good a buy!!"*: 另一种选择是使用一个“密码短语”,可能类似于:“I shop 4 good a buy!!”。本句将 pass phrase 放在引号中,意在模仿 password。实际上,passphrase 在 IT 领域已经成为一个广为接受的词,表示“密码短语”、“口令短语”的意思。

³ *It is difficult to hack, and provides a variety of uppercase, lowercase, and special characters and could be site specific and easy to remember.*: 它很难被黑客破解,而且提供了各种大写、小写和特殊字符,还可能对于网站来说是特定的并易于记忆。

II. Translate the following terms or phrases from English into Chinese and vice versa:

- | | |
|------------------------|----------|
| 1. privacy policy | 5. 零售网站 |
| 2. identity theft | 6. 信用卡 |
| 3. affiliate marketing | 7. 货到付款 |
| 4. postal money order | 8. 安全套接层 |

Glossary

(词汇表)

A

abbreviate /ə'brɪ:vieɪt/ v.	缩写 (1C)
abstract /æb'strækt/ v.	把...抽象出来; 提出, 抽出 (5C)
abstract machine	抽象机 (2C)
abstraction /æb'strækʃən/ n.	抽象; 提取 (5A)
abundance /ə'bʌndəns/ n.	大量, 丰富, 充足 (6C)
abusive /ə'bjʊ:sɪv/ a.	谩骂的; 毁谤的 (11A)
accessory /ək'sesəri/ n.	附件, 配件 (1C)
accommodate /ə'kɒmədeɪt/ v.	容纳; 使适应 (4C)
accountability /ə'kaʊntə'bɪləti/ n.	负有责任 (10A)
accounting /ə'kaʊntɪŋ/ n.	会计 (制度); 记账; 结账; 结算 (12B)
accusation /ækju:'zeɪʃən/ n.	指控, 控告 (11C)
achiever /ə'tʃi:və/ n.	事业成功的人 (11A)
activate /'æktɪveɪt/ v.	激活, 启动 (2C)
actuator /'æktju:ɪtə/ n.	致动器; 传动装置; 执行机构 (9B)
add-on /'ædən/ n.	附加物; 附加装置 (1C)
address /ə'dres/ v.	编址; 寻址 (3C)
Address box	地址框 (8C)
address bus	地址总线 (2A)
addressee /,ædre'si:/ n.	收信人; 收件人 (8B)
addressing /ə'dresɪŋ/ n.	编址; 寻址 (8A)
adjacent /ə'dʒeɪsənt/ a.	相邻的, 毗连的 (2C)
administer /əd'mɪnɪstə/ v.	掌管; 实施 (4C)
administrator /əd'mɪnɪstreɪtə/ n.	(系统、程序等的) 管理员 (6A)
advent /'ædvent/ n.	出现, 到来 (1A)
adverse /'ædvɜ:s, æd've:s/ a.	不利的, 有害的 (2C)
adware /'ædwɛə/ n.	广告软件 (10B)
affiliate /ə'fɪliət/ n.	附属机构; 分公司 (12C)
affiliate marketing	联属网络营销, 联盟营销 (12C)
affiliation /ə'fɪli'eɪʃən/ n.	联系; 从属关系 (12C)
aforementioned /ə'fɔ:menʃənd/ a.	前面提到的 (9A)
aggregate /'ægrɪget/ a.&n.	聚集的; 合计的 (4C) / 总数, 合计; 聚集体 (10A)
aggregation /,ægrɪ'geɪʃən/ n.	聚集, 聚合, 集合 (6C)
agile /'ædʒaɪl/ a.	敏捷的; 灵活的 (9A)
albeit /ɔ:l'bi:t/ conj.	尽管, 即使 (6C)
algorithm /'ælɡərɪðəm/ n.	算法 (1B)
align /ə'leɪn/ v.	对准, 对齐 (11B)
alignment /ə'leɪnmənt/ n.	对准, 对齐 (11B)
allocate /'æləkeɪt/ v.	分配; 分派 (2B)

allot /ə'lɒt/ v.	分配; 分派 (2B)
alternate /ɔ:l'te:nit, 'ɔ:l'te-/ a.	供选择的, 供替换的; 备用的 (7B)
alternatively /ɔ:l'te:nətɪvli/ ad.	或者, 非此即彼 (7B)
ambiguous /æm'bigjuəs/ a.	含糊不清的, 模棱两可的 (3A)
amendment /ə'mendmənt/ n.	修正案, 修正条款 (11C)
analog(ue) /'ænələg/ a.	模拟的 (1A)
analogous /ə'næləgəs/ a.	相似的; 可比拟的 (to/with) (8B)
analogously /ə'næləgəsli/ ad.	相似地, 类似地 (6C)
analogy /ə'nælədʒi/ n.	比拟, 类推, 类比 (5C)
analyst /'ænəlist/ n.	分析员, 分析师 (4B)
analytic(al) /,æne'lɪtɪk(əl)/ a.	分析的 (1A)
Analytical Engine	分析机, 解析机 (1A)
analyzer /'ænəlaɪzə/ n.	分析程序, 分析算法, 分析器 (2C)
animated /'ænɪmeɪtɪd/ a.	动画 (片) 的; 活生生的 (1C)
animation /,æni'meɪʃən/ n.	动画 (制作) (8C)
annotate /'æneuteɪt/ v.	(给...) 作注解 (或评注) (9C)
annotation /,æneuteɪʃən/ n.	注解, 注释, 评注 (9C)
anomaly /ə'nɒməli/ n.	异常 (现象); 不按常规 (6A)
anonymous /ə'nɒnɪməs/ a.	匿名的 (11C)
antenna /æn'tenə/ n.	天线 (9B)
antitrust /,æntɪ'trʌst/ a.	反托拉斯的, 反垄断的 (11A)
antivirus software	防病毒软件 (10B)
appealing /ə'pi:liŋ/ a.	吸引人的; 有感染力的 (3B)
append /ə'pend/ v.	附加 (8B)
applet /'æplət/ n.	小应用程序 (3B)
application /,æplɪ'keɪʃən/ n.	应用程序, 应用软件 (1B)
approximation /ə'prɒksɪ'meɪʃən/ n.	近似 (值) (1A)
arcane /ɑ:'keɪn/ a.	神秘的, 晦涩难解的 (3B)
archaeology /,ɑ:kɪ'ɒlədʒi/ n.	考古学 (6C)
architecture /'ɑ:kɪtektʃə/ n.	体系结构 (1A)
archive /'ɑ:kɑɪv/ n.	[常作~s] 档案; 档案馆, 档案室 (9A)
arena /ə'ri:nə/ n.	竞技场; 活动场所 (8A)
arms race	军备竞赛 (11C)
array /ə'rei/ n.	数组; 一系列 (3A)
artifact /'ɑ:tɪfækt/ n.	人工制品, 制造物 (5B)
artificial intelligence	人工智能 (1A)
assembler /ə'semblə/ n.	汇编程序, 汇编器 (4A)
assembly /ə'sembli/ n.	部件; 组合体 (12A)
assembly code	汇编代码 (3B)
assembly language	汇编语言 (3A)
assignment statement	赋值语句 (3A)
assortment /ə'sɔ:tmənt/ n.	分类; 各种各样 (12B)
assurance /ə'ʃʊərəns/ n.	保证; 把握, 信心 (9B)
asterisk /'æstərɪsk/ n.	星号 (3C)
asynchronous /eɪ'sɪŋkronəs/ a.	不同时的; 异步的, 非同步的 (4C)
athletic /æθ'letɪk/ a.	运动的, 体育的 (12B)
attachment /ə'tætʃmənt/ n.	附件 (10C)
audit /'ɔ:dɪt/ v.	审计, 查账 (9B)
augment /ɔ:g'mənt/ v.	扩大; 增加 (9B)
augmented reality	增强现实 (9B)

authenticate /ɔ:'θentikeɪt/ v.
 authentication /ɔ:'θenti'keɪʃən/ n.
 authenticity /ɔ:'θen'tisiti/ n.
 author /'ɔ:θə/ v.
 authorize /'ɔ:θəraɪz/ v.
 authorized /'ɔ:θəraɪzd/ a.
 automate /'ɔ:təmeɪt/ v.
 automated /'ɔ:təmeɪtɪd/ a.
 automation /ɔ:tə'meɪʃən/ n.
 autonomous /ɔ:'tɒnəməs/ a.
 autonomous agent
 autonomy /ɔ:'tɒnəmi/ n.
 avatar /æ've'tɑ:/ n.
 avoidance /ə'vɔɪdəns/ n.

验证, 鉴别 (11C)
 验证, 鉴别 (4C)
 可靠性, 真实性 (10A)
 著作, 写作; 编写 (11B)
 授权; 委托 (9C)
 经授权的, 特许的 (4C)
 使自动化 (1B)
 自动化的 (1B)
 自动化 (1A)
 自治的, 自主的, 独立的 (4C)
 自主主体 (4C)
 自治 (权), 自主性 (9B)
 化身 (9B)
 避免; 避开; 回避 (9B)

B

backbone /'bækbəʊn/ n.
 backcountry /'bæk.kʌntri/ a.
 backdoor /'bæk'dɔ:/ n.
 backend /'bækend/ a.
 backfire /'bæk'faɪə/ v.
 backup /'bækʌp/ n. & a.
 balance /'bæləns/ n.
 bandwidth /'bændwɪðθ/ n.
 banking /'bæŋkɪŋ/ n.
 banner /'bæne/ n.
 bar /bɑ:/ n.
 bar chart
 bar code
 barcode /'bɑ:kəʊd/ n.
 base ten notation
 baseline /'beɪsləɪn/ n.
 batch /bætʃ/ n.
 battlefield /'bætlfɪ:ld/ n.
 beep /bi:p/ n.
 beneficiary /'benɪ'fɪʃəri/ n.
 billing /'bɪlɪŋ/ n.
 binary /'baɪnəri/ a. & n.
 binary notation
 biological pathway
 biomedical /'baɪəu'medɪkəl/ a.
 biomedicine /'baɪəu'medɪsɪn/ n.
 biometric(al) /'baɪəu'metrɪk(əl)/ a.
 biometrics /'baɪəu'metrɪks/ n.
 bit /bɪt/ n.
 bit pattern
 black box testing
 blackboard model

骨干 (网), 基干 (网) (6C)
 偏僻乡村的 (8C)
 后门 (10C)
 后端的 (9A)
 发生意外, 产生事与愿违的结果 (9C)
 备份, 后备 (2C) / 备份的, 后备的 (6B)
 余额; 差额; 结算 (6B)
 带宽 (4C)
 银行业 (务) (1A)
 标题, 横幅 (11B)
 条; 条形图 (3C)
 条形图 (3C)
 条形码 (12A)
 条形码 (9C)
 以 10 为底的记数法 (8A)
 基线; 基准; 基础 (5B)
 批, 批量, 成批 (9A)
 战场 (9B)
 短促而尖利的声音, 嘟 (9C)
 受益人, 受惠者 (9B)
 开 (账) 单; 记账 (9A)
 二进制的 (1A) / 二进制 (数) (8A)
 二进制记数法 (8A)
 生物学通路 (6C)
 生物医学的 (6C)
 生物医学 (6C)
 生物统计学的 (12A)
 生物统计学 (9B)
 位, 比特 (1A)
 位模式 (2B)
 黑盒测试 (法) (5B)
 黑板法模型 (2C)

blackjack /'blækɔ:dʒæk/ n.	21点(一种牌戏)(11C)
blade server	刀片服务器(1C)
blanket /'blæŋkit/ v.	用毯子(或毯状物)盖(或裹); 覆在...的上面(9C)
block /blɒk/ n.	(字、信息、程序、数据等的)块; 分程序(2A)
blog /blɒg/ n.	博客, 网志, 网络日志(weblog的缩略)(6C)
blooper /'blu:pə/ n.	过失, 失礼(11A)
blueprint /'blu:print/ n.	蓝图(4B)
blurry /'ble:ri/ a.	模糊的, 难辨认的(1C)
bombsight /'bɒmsait/ n.	轰炸瞄准器(1A)
bot /bɒt/ n.	机器人; 僵尸程序(= robot)(10C)
botnet /'bɒtnet/ n.	僵尸网络(10B)
boundary value	[数]边界值(5B)
bounds checking	边界检查(3B)
boxer /'bɒksə/ n.	拳击运动员, 拳师(8C)
bracket /'brækit/ n.	括号(3C)
brand-new /'brænd'nju:/ a.	全新的, 崭新的(10B)
breach /bri:tʃ/ n.	破坏; 违反(10A)
bridge /brɪdʒ/ n.	网桥, 桥接器(4C)
broadband /'brɔ:dbænd/ a.	宽带的(8A)
broker /'brəʊkə/ n.	代理者; 代理程序(3B)
browse /braʊz/ v.	浏览(3B)
browser /'braʊzə/ n.	浏览器(2C)
bug /bʌg/ n.	(程序)错误, 故障(4A)
buggy /'bʌgi/ a.	有(程序)错误的, 有故障的(6C)
building block	积木块, 构建模块, 构件(4B)
built-in /'bɪlt'in/ a.	内置的, 内部的(1A)
bulky /'bʌlki/ a.	体积大的, 庞大的; 又大又笨的(6C)
bureaucratic /'bjʊərəʊ'krætɪk/ a.	官僚(政治)的; 官僚主义的(4B)
bus /bʌs/ n.	总线(1A)
bus topology	总线拓扑结构(7A)
bust /bʌst/ n.	崩溃, 不景气(12A)
bypass /'baɪpɑ:s/ v.	绕过; 越过(12C)
byte /baɪt/ n.	字节(1A)
bytecode /'baɪtkəʊd/ n.	字节码(3B)

C

cable television	有线电视(8A)
cabling /'keɪblɪŋ/ n.	电缆(9C)
calculation /,kælkju'leɪʃən/ n.	计算(1A)
calculator program	计算(器)程序(10C)
callout /'kɔ:laut/ n.	标注(9A)
callout box	标注框(9A)
cancellation /,kænsə'leɪʃən/ n.	取消; 删去(6B)
capitalization /,kæpɪtəlaɪ'zeɪʃən; -lɪz/ n.	大写字母的使用(8C)
captivating /'kæptɪveɪtɪŋ/ a.	迷人的, 可爱的(8C)
cardinality /,kɑ:di'nælɪti/ n.	基数性; (集的)势, (集的)基数(6A)
carnivore /'kɑ:nɪvɔ:/ n.	食肉动物(11C)
carrier /'kæriə/ n.	载波(7A) / 运输公司; 运输工具; 通信公司(9C)

carrier sense	载波检测, 载波监听 (7A)
cartridge /'kɑ:trɪdʒ/ n.	盒, 匣 (4A)
carve /kɑ:v/ v.	雕刻, 刻 (9A)
cascade /kæ'skeɪd/ n.	小瀑布; 瀑布状物; 级联; 层叠 (5A)
cascading /kæ'skeɪdɪŋ/ a.	级联的; 层叠的 (6B)
cascading rollback	级联回滚 (6B)
cash on delivery	货到付款, 交货付款 (12C)
cash register	现金出纳机, 收银机 (9C)
casino /kə'si:nəʊ/ n.	(有表演、舞池等的) 卡西诺赌场 (11C)
categorize /'kætɪgəraɪz/ v.	将...分类 (1C)
cater /'keɪtə/ v.	满足需要; 迎合; 考虑 (<i>for, to</i>) (5A)
cathode /'kæθəʊd/ n.	阴极 (2A)
cathode ray tube	阴极射线管 (2A)
cell phone	蜂窝电话, 移动电话, 手机 (1C)
cellular /'seljələ/ a.	蜂窝状的, 多孔的 (6C)
cellular phone	蜂窝电话, 移动电话, 手机 (6C)
censor /'sensə/ v.	审查, 检查 (11C)
census /'sensəs/ n.	人口普查 (1A)
central processing unit	中央处理器 (1A)
centrifugal /sen'trɪfjuɡəl/ a.	离心的 (12C)
centrifugal pump	离心泵 (12C)
chain-letter /'tʃeɪn.leɪtə/ v.	向...发送连锁信 (或连锁邮件) (11A)
challenging /'tʃælɪndʒɪŋ/ a.	挑战性的 (6C)
chatty /'tʃæti/ a.	聊天式的, 轻松而亲切的; 爱闲聊的 (11A)
check box	复选框, 选择框, 校验框 (4B)
checkers /'tʃekəz/ n.	(美) 西洋跳棋 (7A)
checkout /'tʃekaut/ n.	结账 (离去); 付款台 (9B)
chip /tʃɪp/ n.	芯片 (1A)
circuitry /'sə:kɪtri/ n.	电路 (1A)
circumvent /,sə:kəm'vent/ v.	绕过; 规避 (5A)
civic-minded /'sɪvɪk'maɪndɪd/ a.	关心公益的; 热心公民 (或市民) 事务的 (10B)
clarity /'klærɪti/ n.	清晰, 明晰 (3B)
class hierarchy	类层次 (3B)
classified /'klæsɪfaɪd/ a.	归入密级的, 保密的 (11A)
click /kɪk/ v. & n.	(鼠标) 单击 (2B) / (鼠标的) 点击 (12B)
client /'klaɪənt/ n.	客户机; 客户程序 (1C)
clip /kɪp/ n.	剪下来的东西; 电影 (或电视) 片段 (2C)
cloud computing	云计算 (6C)
cluster /'klʌstə/ n.	(人或物的) 群, 组; (果实、花等的) 串, 束, 簇 (7A)
clutter /'klʌtə/ n.	凌乱, 杂乱; 杂乱的东西 (3B)
coaxial /,kəʊ'æksɪəl/ a.	同轴的 (7B)
coaxial cable	同轴电缆 (7B)
coexist /,kəʊɪg'zɪst/ v.	共存; 同时存在 (4C)
cognition /kɒɡ'nɪʃən/ n.	认识, 认知 (1B)
coherent /kəʊ'hɪərənt/ a.	相干的; 一致的; 协调的 (6A)
coincide /,kəʊɪn'saɪd/ v.	相符, 相一致; 重合 (8B)
cold chain	冷链 (9B)
collaboration /kə'læbə'reɪʃən/ n.	合作; 协作 (1B)
collaborative /kə'læbə'reɪtɪv/ a.	合作的, 协作的 (4C)

collide /kə'laɪd/ v.	冲突; 碰撞 (7C)
collusion /kə'lu:ʒən/ n.	共谋, 勾结, 串通 (10B)
colon /'kəʊlən/ n.	冒号 (8C)
comic strip	(报刊上的) 连环漫画 (9C)
command line	命令行 (9B)
commit /kə'mɪt/ v.	提交, 委托; 承诺 (6B)
commit point	提交点 (6B)
compact disc	光盘 (2A)
compatible /kəm'pæteɪbl/ a.	兼容的 (2B)
compilation /,kəm'pi:leɪʃən/ n.	编译; 汇编 (2C)
compile /kəm'paɪl/ v.	汇编; 编译 (1A)
compiled code	编译执行的代码 (3B)
compiled language	编译执行的语言 (4A)
compiler /kəm'paɪlə/ n.	编译程序, 编译器 (3A)
complexity /kəm'pleksɪti/ n.	复杂性, 错综性 (1B)
complication /,kəm'pli:keɪʃən/ n.	复杂情况; 困难, 难题 (12B)
comprehend /,kəm'pri'hend/ v.	理解, 领会 (1B)
compression /kəm'preʃən/ n.	压缩 (2C)
compromise /'kɒmp'remaɪz/ v.	连累; 危及, 损害 (5C)
computational /,kəm'pjʊ:'teɪʃənəl/ a.	计算(机)的 (1B)
compute-intensive /kəm'pjʊ:'tɪn'tensɪv/ a.	计算密集型的 (1C)
computer vision	计算机视觉 (1B)
computer-aided design	计算机辅助设计 (1C)
conceive /kən'si:v/ v.	(构)想出 (1A)
conceptual /kən'septʃuəl/ a.	概念的 (6A)
conceptually /kən'septʃuəli/ ad.	概念上 (1A)
concise /kən'saɪs/ a.	简明的, 简要的 (11A)
concurrency /kən'kʌrənsi/ n.	同时发生, 并发, 并行性 (4C)
concurrent /kən'kʌrənt/ a.	同时发生的, 并发的, 并行的 (2B)
conditional statement	条件语句 (3A)
conducive /kən'dju:sɪv/ a.	有助的, 有益的 (to) (8A)
confidential /,kən'fɪ'denʃəl/ a.	秘密的, 机密的 (6C)
confidentiality /'kɒn'fɪ'denʃi'ælɪti/ n.	机密性 (10A)
configuration /kən'fɪgju'reɪʃən/ n.	配置 (1C)
configuration item	配置项 (2C)
configure /kən'fɪg/ v.	配置 (1C)
congest /kən'dʒest/ v.	拥挤; 拥塞 (9C)
conjecture /kən'dʒektʃə/ n.	推测, 猜想 (1A)
connectivity /kən'ek'tɪvɪti/ n.	连通(性), 连接(性) (4C)
consecutive /kən'sekjʊtɪv/ a.	连续的 (5B)
console /'kɒnsəʊl/ n.	控制台, 操纵台 (1C)
consolidate /kən'sɒlɪdeɪt/ v.	(把...) 联为一体, 合并; 巩固 (4B)
consortium /kən'sɔ:tɪəm/ n.	([复] -tia / -tɪə / 或 -tiums) 会, 社; (国际性的) 财团 (4C)
constrain /kən'streɪn/ v.	约束, 限制 (3B)
constraint /kən'streɪnt/ n.	约束, 限制 (5A)
consumer electronics	消费电子产品 (9C)
contagion /kən'teɪdʒən/ n.	(接)触(传)染 (10C)
contention /kən'tenʃən/ n.	争用; 争夺 (4C)
context switch	上下文转换, 语境转换 (2B)

controller /kən'trəʊl/ n.	控制器 (2B)
convergence /kən'veɪdʒəns/ n.	会聚; 结合; 收敛 (1C)
conversion /kən'veɪʃən/ n.	转换; 转变; 兑换 (12A)
converter /kən'veɪtə/ n.	转换器, 转换程序 (7C)
convoluted /'kɒnvəlu:tɪd/ a.	盘绕的; 盘错的, 错综复杂的 (3B)
cookie /'kuki/ n.	小甜饼 (指一种临时保存网络用户信息的结构) (11C)
cooperative /kəu'ɒpəreɪtɪv/ a.	合作的, 协作的 (9B)
Copy command	复制命令 (8C)
corporate /'kɔ:pəɪt/ a.	公司的; 社团的 (4C)
corrupt /kə'rʌpt/ v.	破坏, 损坏; 腐蚀 (5A)
corruption /kə'rʌpʃən/ n.	破坏; 腐化; 讹误 (6B)
cosmetic /kɒz'metɪk/ a.	化妆用的; 装饰性的; 非实质性的 (8C)
cost effectiveness	成本效益 (4C)
cost-effective /'kɒsti'fektɪv/ a.	有成本效益的; 合算的 (5A)
counteract /,kaunte'rækt/ v.	对...起反作用; 对抗; 抵消 (5B)
countermeasure /'kaunte'meʒə/ n.	对策, 对抗手段 (10A)
counterpart /'kauntepɑ:t/ n.	对应的物 (或人) (4A)
counterterrorism /'kaunte'terəɪzəm/ n.	反恐怖主义, 反恐 (6C)
country code	国家代码 (8A)
coupon /'ku:pən/ n.	购物优惠券, 礼券, 赠券 (12B)
courier /'kuriə/ n.	信使 (11A)
coverage /'kʌvərɪdʒ/ n.	覆盖 (范围) (5B)
coworker /kəu'wɜ:kə/ n.	同事 (11A)
crack /kræk/ v.	破译 (1A)
credit card	信用卡 (9C)
criterion /kraɪ'tiəriən/ n.	〔复〕-ria/-riə/或-rions) 标准, 准则 (1C)
critter /'krite/ n.	〈美口〉生物; 动物 (11A)
cryptic /'kriptɪk/ a.	隐秘的; 令人困惑的 (3B)
culprit /'kʌlpɪt/ n.	罪犯; 被控犯罪的人 (11C)
cursor /'kɜ:sə/ n.	光标 (2A)
customarily /'kʌstəməɪli; ,kʌstə'merɪli/ ad.	习惯上, 通常 (1C)
customization /kʌstəmaɪ'zeɪʃən/ n.	定制, 用户化 (9A)
customize /'kʌstəmaɪz/ v.	定制, 使用户化 (2B)
customized /'kʌstəmaɪzd/ a.	定制的, 用户化的 (12A)
cyber /'saɪbə/ a.	计算机 (网络) 的 (8A)
cyber cafe	网吧 (8A)
cyber-physical /,saɪbə'fɪzɪkəl/ a.	信息物理的 (6C)
cyberspace /'saɪbəspeɪs/ n.	计算机空间, 网络空间 (8C)

D

data bus	数据总线 (2A)
data capture	数据捕获, 数据收集 (10B)
data coloring	数据着色 (9A)
data declaration	数据声明 (3A)
data flow diagram	数据流程图 (4B)
data item	数据项 (6B)
data link	数据链路 (7C)
data logging	数据 (事件) 记录; 数据登录 (9B)

data mining	数据挖掘 (6C)
data pattern	数据模式 (6C)
data point	数据点 (1C)
data set	数据集 (6C)
data stream	数据流 (6C)
data warehouse	数据仓库 (6C)
database architecture	数据库体系结构 (6A)
database management system	数据库管理系统 (6A)
dated /'deɪtɪd/ a.	过时的 (11B)
datum /'deɪtəm/ n.	([复] data) 数据 (1A)
de facto /di:'fæktəu/	(拉) 实际的, 事实上的 (3B)
deadlock /'dedlɒk/ n.	死锁; 僵局 (6B)
dealer /'di:lə/ n.	发牌者; 商人 (11C)
debit /'deɪt/ n.	借方; 借记, 借入 (9C)
debit card	借记卡, 借方卡 (9C)
debug /di:'bʌg/ v.	调试, 排除 (程序) 中的错误 (4A)
debugger /di:'bʌgə/ n.	调试程序, 排错程序 (4A)
decimal /'desɪməl/ a.	十进制的 (1A)
decimal notation	十进制记数法 (8A)
deck /dek/ n.	卡片叠, 卡片组 (1A)
decode /di:'kəʊd/ v.	译 (码), 解 (码) (2B)
decompression /di:kəm'preʃən/ n.	解压缩 (2C)
decorative /'dekəreɪtɪv/ a.	装饰性的 (11B)
decrement /'dekri:mənt/ v.	减少, 减缩 (6B)
dedicated /'dedɪkeɪtɪd/ a.	专用的 (1A)
deduce /dɪ'dʒu:s/ v.	推论, 推断 (3A)
deem /'di:m/ v.	认为, 视为 (5B)
default /dɪ'fɔ:lt/ n.	默认, 缺省, 系统设定值 (4A)
defective /dɪ'fektɪv/ a.	有缺点的, 有缺陷的, 有毛病的 (11C)
definitive /dɪ'fɪnɪtɪv/ a.	决定性的; 确定的; 规定的 (5A)
degradation /,degreɪ'deɪʃən/ n.	降级, 退化 (7C)
degrade /dɪ'greɪd/ v.	使降级, 使退化 (10A)
degraded /dɪ'greɪdɪd/ a.	降级的, 退化的 (7B)
déjà vu /,deɪʒə:'vu:/	(法) 似曾经历的错觉 (5C)
deliverable /dɪ'lɪvərəbəl/ n.	[常作复] 可交付使用的产品 (4B)
delivery on payment	付款交货 (12C)
delve /delv/ v.	搜索, 翻查 (3B)
demodulate /di:'mɒdjuleɪt/ v.	解调 (9B)
demodulator /di:'mɒdjuleɪtə/ n.	解调器 (2A)
denote /dɪ'noʊt/ v.	表示; 意思是 (6A)
dependency /dɪ'pendənsi/ n.	依靠, 依赖 (2C)
deploy /dɪ'plɔɪ/ v.	部署 (12C)
deployment /dɪ'plɔɪmənt/ n.	部署; 展开 (9A)
derivation /,derɪ'veɪʃən/ n.	派生 (物) (8C)
descriptor /dɪ'skriptə/ n.	描述符, 解说符 (2B)
desensitize /,di:'sensɪtaɪz/ v.	使不敏感, 使降低敏感性 (10B)
designate /'deɪzɪneɪt/ v.	指定, 指明; 命名; 指派 (3C)
desk /desk/ n.	服务台; 部门 (8C)
deskside /'desksaɪd/ a.	桌边 (型) 的 (9A)
desktop /'deskɒp/ a. & n.	桌面的; 台式 (计算机) 的 (1A) / 台式

destine /'destin/ v.
deterministic /di,tə:'mi'nistik/ a.
deterrence /di'terəns/ n.
detrimental /,detrɪ'mentəl/ a.
devastating /'devəsteɪtɪŋ/ a.
device driver
dexterity /dek'sterɪti/ n.
diagnostic /,daɪə'gnɒstɪk/ a.
dialog box
dial-up /'daɪəlʌp/ a.
Difference Engine
differentiate /,dɪfə'renʃieɪt/ v.
diffusion /dɪ'fju:ʒən/ n.
digit /'dɪdʒɪt/ n.
digitize /'dɪdʒɪtaɪz/ v.
dinosaur /'daɪnəsɔ:/ n.
dip /dɪp/ n.
direct debit
directory /dɪ'rektəri/ n.
directory path
disability /,dɪsə'bɪlɪti/ n.
disastrous /dɪ'zɑ:stəs/ a.
disclose /dɪs'kleʊz/ v.
disclosure /dɪs'kleʊʒə/ n.
discretion /dɪs'kreʃən/ n.
discrimination /dɪs,krɪmɪ'neɪʃən/ n.
disk array
disk drive
dismantle /dɪs'mæntl/ v.
disparate /'dɪspəɪt/ a.
dispatcher /dɪs'pætʃə/ n.
disrupt /dɪs'rʌpt/ v.
disruption /dɪs'rʌpʃən/ n.
disruptive /dɪs'rʌptɪv/ a.
distorted /dɪs'to:tɪd/ a.
distributed /dɪ'strɪbʊtɪd/ a.
distributor /dɪ'strɪbjʊtə/ n.
documentation /,dɒkjʊmen'teɪʃən/ n.

domain /də'meɪn/ n.
domain name system
domain name
doorway /'dɔ:weɪ/ n.
dormant /'dɔ:mənt/ a.
dot-bomb /'dɒtbɒm/ n.
dot-com /'dɒtkɒm/ n.
dot-matrix printer
dotted decimal notation
double-check /'dʌblɪ'tʃek/ v.

(计算)机; 桌面 (9A)
预定, 指定 (往某一目的地或供某种用途) (*for*) (8B)
决定论的; 确定性的 (9B)
威慑 (10A)
有害的; 不利的 (1C)
破坏性极大的, 毁灭性的 (6B)
设备驱动程序 (2B)
灵巧, 敏捷 (1B)
诊断的 (9C)
对话框 (3B)
拨号的 (8A)
差分机 (1A)
区分, 区别 (1C)
扩散 (9B)
数字 (1A)
使 (数据) 数字化 (2C)
恐龙; (尤指废弃过时的) 庞然大物 (2B)
(暂时或小幅度的) 减少; 下降 (12A)
直接借记, 直接付款 (12C)
目录 (2B)
目录路径 (2B)
无能力; 残疾 (1B)
灾难性的 (6B)
使显露; 泄露, 透露 (6C)
泄露, 透露 (10A)
斟酌决定 (或处理) 的自由 (2B)
区别; 歧视 (11A)
磁盘阵列 (9A)
磁盘机, 磁盘驱动器 (2A)
解散; 拆开; 废除 (7A)
完全 (或根本) 不同的; 不相干的 (12A)
调度程序 (2B) / (车辆) 调度员 (9C)
扰乱; 使中断 (7B)
扰乱; 中断 (10A)
扰乱性的, 制造混乱的, 捣乱的 (10C)
扭曲的, 变形的 (11C)
分布 (式) 的 (2C)
经销商; 批发商 (12C)
文件编制, 文档编制; [总称] 文件证据, 文献资料 (4B)
领域, 域 (4C)
域名系统 (8A)
域名 (8A)
出入口, 门口 (8C)
休眠的; 暂停活动 (或作用) 的 (10C)
网络炸弹 (12A)
网络公司 (12A)
点阵打印机 (2A)
点分十进制记数法 (8A)
复核; 从两方面查对 (10B)

down /daun/ v.	击倒; 击落; 倒下 (7B)
downright /'daunrait/ ad.	彻底地, 完全地 (11C)
downturn /'dauntə:n/ n.	衰退; 下降趋势 (12A)
dredge /dredʒ/ v.	疏浚; 挖掘; 采捞 (6C)
drive /draiv/ n.	驱动器 (2A)
driver /'draivə/ n.	驱动程序, 驱动器 (2B)
drone /dreun/ n.	游手好闲者, 寄生虫 (10C)
drop shipping	直达货运 (12C)
drop-down menu	下拉式菜单 (或选项屏) (4B)
droplet /'drɒplit/ n.	小滴 (2A)
drum /drʌm/ n.	磁鼓 (2A)
duct /dʌkt/ n.	管道; 导管 (9C)
duke /djʊ:k/ n.	公爵 (3B)
duplicate /'djʊ:plikeit/ v.	复制; 重复 (8C)
duplicate /'djʊ:plikit/ a.	复制的; 副 (本) 的; 重复的 (7A)

E

eavesdrop /'i:vzdrɒp/ v.	偷听, 窃听 (<i>on</i>) (10A)
editor /'edite/ n.	编辑程序, 编辑器 (2C)
e-label /'i:leibəl/ n.	电子标签 (9B)
elastic /i'læstik/ a.	有弹性的; 可伸缩的 (9A)
electrical contact	电触点 (1A)
electronic book reader	电子 (图) 书阅读器 (9C)
electronic picture frame	电子相框 (9C)
embed /im'bed/ v.	把...嵌入 (1A)
embedded /im'bedid/ a.	嵌入 (式) 的 (1C)
emoticon /i'məutikɒn/ n.	情感符 (<i>emotion icon</i> 的缩合) (11A)
emulate /'emjuleit/ v.	模拟, 仿真, 模仿 (10B)
encapsulation /in,kæpsju'leɪʃən; -sə-/ n.	封装 (3B)
encipher /in'saɪfə/ v.	把...译成密码 (1A)
encode /en'keud/ v.	把...编码; 把...译成电码 (或密码) (4A)
encompass /in'kæmpəs/ v.	包含, 包括 (4C)
encrypt /in'kript/ v.	把...加密 (10B)
encryption /in'kripʃən/ n.	加密 (4C)
encryption key	(加密) 密钥 (10A)
end user	最终用户, 终端用户 (6A)
endpoint /'endpoint/ n.	端点 (7B)
energy harvesting	能量收集 (9B)
enforcement /in'fɔ:smənt/ n.	实施; 强制执行 (1C)
entail /in'teɪl/ v.	使成为必要, 需要 (9B)
Enter key	回车键 (8C)
entirety /in'taiəti/ n.	全部; 整体 (6B)
entity /'entiti/ n.	实体 (4B)
entity relationship diagram	实体关系图, 实体联系图, E-R 图 (4B)
entity-relationship model	实体关系模型 (6A)
envision /in'viʒən/ v.	想象; 设想 (9A)
equivalence /i'kwivələns/ n.	等价; 相等 (5B)
equivalence partitioning	等价 (类) 划分 (5B)

eradication /i,rædi'keifən/ n.	根除; 消灭 (10C)
erase /i'reiz/ v.	擦除, 消除, 清除 (2A)
erroneous /i'reunies/ a.	错误的, 不正确的 (6B)
essence /'esəns/ n.	本质, 实质 (3A)
etch /etʃ/ v.	蚀刻 (1A)
Ethernet /'i:θə.net/ n.	以太网 (标准) (4C)
ethical /'eθikəl/ a.	道德的; 伦理的 (11C)
etiquette /'etiket/ n.	礼节; (行业中的) 道德规范; 规矩 (11A)
eventual /i'ventʃuəl/ a.	最终发生的, 最后的 (6C)
evoke /i'vəuk/ v.	唤起; 使人想起 (12C)
excerpt /ək'se:pt/ v.	摘录; 引用 (11A)
exclusive lock	排它锁, 互斥 (型) 锁 (6B)
executable /'eksikju:təbl/ a. & n.	可执行的 (3A) / 可执行文件 (10B)
execution /.eksɪ'kjʊʃən/ n.	执行, 运行 (1A)
execution path	执行路径 (5B)
exercising test	压力测试 (5B)
exhaustive /ig'zɔ:stiv/ a.	全面而彻底的, 详尽无遗的 (5B)
existing /ig'zistɪŋ/ a.	现存的, 现有的, 现行的 (1B)
expedite /'ekspidaɪt/ v.	迅速执行; 加速 (9B)
expiration /.ekspraɪ'reɪʃən/ n.	满期, 届期 (9C)
expire /ik'spaɪə/ v.	满期, 届期 (9C)
exploratory /ik'splɒrətəri/ a.	探索的; 勘探的 (5A)
expression /ik'spreʃən/ n.	表达式 (3A)
extensibility /ik.stense'biliti/ n.	可扩展性, 可扩充性 (4C)
extension /ik'stenʃən/ n.	扩展, 扩充; 扩展名 (2B)
extract /ik'strækt/ v.	提取, 抽取, 析取 (6C)
extraction /ik'strækʃən/ n.	提取, 抽取, 析取 (6C)

F

fabricate /'fæbrɪkeɪt/ v.	制作 (1A)
factor /'fæktə/ v.	把...分解成 (into) (5C)
familiarize /fə'milləraɪz/ v.	使熟悉; 使通晓 (7B)
fault isolation	故障隔离 (10A)
fault tolerance	容错 (4C)
ferry /'feri/ n. & v.	渡船; 摆渡; 渡口 / 渡运; 运送 (8C)
fictional /'fɪkʃənəl/ a.	小说的; 虚构的 (2B)
field /fi:ld/ v. & n.	派...上场; 实施 (3B) / 字段; 域; 信息组 (6A)
file server	文件服务器 (7A)
filestore /'faɪlstɔ:/ n.	文件存储 (器) (2C)
film clip	剪片 (2C)
fingerprint /'fɪŋɡeɪntrɪnt/ n.	指纹 (印), 手印 (12A)
fingerprint reader	指纹读取器 (12A)
firewall /'faɪəwɔ:l/ n.	防火墙 (4C)
firmware /'fɜ:mweə/ n.	[总称] 固件 (2A)
fitting /'fɪtɪŋ/ n.	试穿, 试衣 (12C)
flag /flæg/ v.	用标记表明 (10B)
flame /fleɪm/ v.	(向...) 发送争论 (或争辩) 邮件 (11A)
flaming /'fleɪmɪŋ/ n.	争论 (特指在邮件讨论组或网络论坛中争论) (11A)

flat file	平面文件, 展开文件 (6A)
flatbed scanner	平板扫描仪 (2A)
flaw /flɔ:/ n.	缺点, 瑕疵 (3B)
flawed /flɔ:d/ a.	有缺点的, 有瑕疵的 (5C)
floppy /'flɒpi/ a.	(松) 软的 (2A)
floppy disk	软 (磁) 盘 (2A)
flowchart /'fləʊtʃɑ:t/ n.	流程图 (3C)
flux /flʌks/ n.	(不断的) 变动; 波动 (6B)
focal /'fəʊkəl/ a.	焦点的 (7A)
focal point	焦点, 活动 (或注意、兴趣等的) 中心 (7A)
foil /fɔil/ v.	挫败; 使受挫折 (12C)
folder /'fəʊldə/ n.	文件夹 (2B)
font /fɒnt/ n.	字体 (5B)
footwear /'fʊtwɛə/ n.	[总称] 鞋类 (12B)
forensic /fə'rensik/ a.	(用于) 法庭的; 法医的 (10A)
forfeit /'fɔ:fit/ v.	丧失, 失去 (5B)
form factor	(电子产品等的) 物理尺寸和形状; 规格 (1C)
formulate /'fɔ:mjuleit/ v.	构想出; 系统地阐述 (8C)
forward slash	正斜杠 (8C)
frame /freim/ n.	帧, 画面 (2C) / 图文框; 框架 (4B)
fraudulent /'frɔ:dʒələnt/ a.	欺骗性的, 欺诈性的 (12C)
fraught /frɔ:t/ a.	[一般作表语] 充满的 (with) (12A)
frequency array	频率数组 (3C)
frequent /'fri:kwent/ v.	常到, 常去, 时常出入于 (11C)
frivolous /'frivələs/ a.	轻薄的; 琐屑的 (11A)
frontend /'frʌnt'end/ n.	前端 (9A)
full-featured /'fʊl'fi:tʃəd/ a.	全特征的, 全功能的 (1C)
function key	功能键, 操作键, 函数键 (2A)
function statement	函数语句 (3A)
functional language	函数式语言 (3A)
functionality /fʌŋkʃə'næliti/ n.	功能性 (2B)
fuse /fju:z/ v.	熔合; 熔凝 (2A)
fusion /'fju:ʒən/ n.	熔合; 融合 (9B)

G

gaggle /'gægl/ n.	(紊乱而有联系的) 一堆 (3B)
gas guzzler	耗油量大的汽车, 油老虎 (9C)
gateway /'geitwei/ n.	网关 (7C)
gauge /geidʒ/ v.	估计, 判断; 计量 (4B)
generator /'dʒenəreɪtə/ n.	生成程序, 生成器 (2C)
generic /dʒi'nerik/ a.	类属的; 一般的; 通用的 (2B)
geocaching /.dʒi:əu'kæʃɪŋ/ n.	地理藏宝 (9B)
geolocation /.dʒi:əuleɪ'keɪʃən/ n.	地理位置, 地理定位 (9B)
geotagging /.dʒi:əu'tæɡɪŋ/ n.	地理标记, 地理标签 (9B)
gift card	礼物卡, 打折卡 (12C)
gigantic /dʒaɪ'ɡæntɪk/ a.	巨大的, 庞大的 (6C)
good-sized /'ɡʊd'saɪzd/ a.	(相当) 大的; 宽大的 (1C)
grammatical /grə'mætɪkəl/ a.	(符合) 语法的 (3A)

granularity /ˌɡrænjuˈlærɪti/ n.
 graphic(al) /ˈɡræfɪk(əl)/ a.
 graphical user interface
 graphics /ˈɡræfɪks/ n.
 graphics card
 green /ɡriːn/ n.
 greeting card
 grid /ɡrɪd/ n.
 grim /ɡrɪm/ a.
 groundbreaking /ˈɡraʊndbreɪkɪŋ/ a.
 gullible /ˈɡʌlibəl/ a.
 guzzler /ˈɡʌzlə/ n.

(颗)粒度(5C)
 图形的, 图示的, 图解的(1B)
 图形用户界面(2B)
 图形, 图形显示(1A)
 显(示)卡(10B)
 (高尔夫)球穴区; 高尔夫球场(8C)
 贺卡(8C)
 网格, 格网(9A) / 系统网络(指输电线路、广播电台、电视台、天然气管道等网络)(9B)
 严厉的; 阴森的(11A)
 开辟新天地的, 开拓性的(10C)
 易受骗的, 易上当的(11A)
 狂饮者; 滥吃者; 大量消耗者(9C)

H

habitat /ˈhæbɪtæt/ n.
 hack /hæk/ v.
 handheld computer
 hand-held scanner
 handle /ˈhændl/ n.
 handler /ˈhændlə/ n.
 handshaking /ˈhændʃeɪkɪŋ/ n.
 haphazard /ˌhæpˈhæzəd/ a.
 harass /ˈhærəs/ v.
 harassment /ˈhærəsment, həˈræs-/ n.
 hard disk
 hard drive
 header /ˈhedə/ n.
 heading /ˈhedɪŋ/ n.
 head-up display
 heterogeneity /ˌhetərəʊdʒiˈniːəti/ n.
 heterogeneous /ˌhetərəʊdʒiːniəs/ a.
 heuristic /hjuˈrɪstɪk/ a.
 hierarchical /ˌhaɪəˈrɑːkɪkəl/ a.
 hierarchy /ˌhaɪəˈrɑːki/ n.
 high-dimensional microarray
 high-rise /ˈhaɪˈraɪz/ a.
 hinge /hɪndʒ/ v.
 histogram /ˈhɪstəʊɡræm/ n.
 hoax /heʊks/ n.
 home page
 homepage /ˈhəʊmpeɪdʒ/ n.
 homogeneous /ˌhəməʊdʒiːniəs/ a.
 hop /hɒp/ v.
 host /heʊst/ n. & v.
 host address
 host program

(动植物的)自然生存环境, 栖息地(9B)
 非法闯入(计算机网络), 黑客攻击(9C)
 手持式计算机, 掌上电脑(1C)
 手持式扫描仪(2A)
 句柄; (文件、对象等的)(名)称(编)号, 名(字编)号(5C)
 处理程序, 处理器(2B)
 握手, 信号交换(2B)
 无计划的, 随意的(8A)
 骚扰; 烦扰(11A)
 骚扰; 烦扰(11A)
 硬(磁)盘(1C)
 硬盘驱动器(2A)
 标题, 报头, 头标; 页眉(3B)
 航向(1B)
 前导显示器, 平视显示器(1B)
 各种各样, 异构性, 非均匀性(4C)
 不同种类的, 多机种的, 异构的(4C)
 试探的, 启发式的(10B)
 分级的, 分层的, 层次的(2B)
 层次, 分层(结构), 分级(结构)(3A)
 高维微数组(6C)
 (建筑)高层的(9B)
 依…而定, 以…为转移(*on, upon*)(11A)
 直方图, 矩形图; 频率分布图(3C)
 骗局; 恶作剧(11A)
 主页(8C)
 主页(11B)
 同种类的; 同性质的; 同构的(4C)
 跳跃(8B)
 主机(8A) / 作…的主机(8C)
 主机地址(8A)
 主机程序, 宿主程序, 主程序(10C)

hosting /'həʊstɪŋ/ n.	托管 (12B)
hotspot /'hɒtspɒt/ n.	(网络) 热点, 热区 (8A)
howl /haʊl/ v.	吼叫; 怒吼 (9C)
HTML tag	HTML 标记 (8C)
hub /hʌb/ n.	(网络) 集线器; (轮) 毂; (兴趣、活动等的) 中心 (7B)
Hungarian /hʌŋ'ɡɜəriən/ a.	匈牙利的 (1A)
hurricane /'hʌrɪkən/ n.	飓风 (8C)
hybrid /'haɪbrɪd/ n. & a.	混合物 (7B) / 混合的 (9A)
hydroelectric /,haɪdrəʊ'lektrɪk/ a.	水力发电的; (电) 由水力发的 (9A)
hyperlink /'haɪpəlɪŋk/ n.	超 (级) 链接 (8C)
hypertext /'haɪpətɛkst/ n.	超 (级) 文本 (4C)
Hypertext Markup Language	超 (级) 文本标记语言 (8C)
Hypertext Transfer Protocol	超 (级) 文本传送 (或传输) 协议 (4C)

I

icon /'aɪkɒn/ n.	图标, 图符 (2B)
identifier /aɪ'dentɪfaɪə/ n.	标识符 (8A)
identity theft	身份 (信息) 盗取 (12C)
imaging /'ɪmɪdʒɪŋ/ n.	成像 (技术) (1C)
immigration /,ɪmɪ'ɡreɪʃən/ n.	移民, 移居 (8C)
impersonate /ɪm'pɜ:səneɪt/ v.	扮演; 模仿; 假冒 (10A)
implant /ɪm'plɑ:nt; -'plænt/ v.	植入, 埋置 (9C)
implementation /,ɪmplɪ'men'teɪʃən/ n.	实现, 执行 (1B)
inadvertent /,ɪnəd'vɜ:tənt/ a.	因疏忽造成的; 粗心大意的 (6B)
inclined /ɪn'klaɪnd/ a.	倾向于...的; 有...爱好 (或天赋) 的 (10B)
incoming /ɪn,kʌmɪŋ/ a.	进来的; 输入的 (8B)
inconsistency /,ɪnkən'sɪstənsɪ/ n.	不一致 (2C)
incorporate /ɪn'kɔ:pəreɪt/ v.	包含, 吸收; 把...合并, 使并入 (1A)
increment /ɪnkrɪ'ment/ v.	增加, 增长 (6B)
incremental /,ɪnkrɪ'mentəl/ a.	增量的; 递增的 (2C)
indefinite /ɪn'defɪnɪt/ a.	不确定的, 不定的 (5B)
indexing /'ɪndeksɪŋ/ n.	编索引; 标引; 加下标; 变址 (3C)
indicative /ɪn'dɪkətɪv/ a.	标示的; 指示的 (9B)
indiscreet /,ɪndɪ'skri:t/ a.	不慎重的, 轻率的 (11A)
inference engine	推理机 (4A)
info /'ɪnfəʊ/ n.	〈口〉信息 (= information) (2C)
information superhighway	信息高速公路 (11A)
infrastructure /'ɪnfə'strʌktʃə/ n.	基础结构 (4C)
infringe /ɪn'frɪndʒ/ v.	违犯, 侵犯 (权利等) (11C)
infringement /ɪn'frɪndʒmənt/ n.	(对他人权利等的) 侵犯, 侵害 (11C)
inheritance /ɪn'herɪtəns/ n.	继承 (性) (3B)
inheritance hierarchy	继承层次 (5C)
inhibit /ɪn'hɪbɪt/ v.	抑制, 约束 (10A)
in-house /ɪn'haʊs/ ad. & a.	在机构内部; 无外援地 / 内部的, 自用的 (12B)
initiate /ɪ'nɪʃɪeɪt/ v.	开始; 发起 (2B)
inject /ɪn'dʒekt/ v.	注射; 注入; 引入 (10C)
ink cartridge	墨盒 (4A)

inkjet /'ɪŋkdʒet/ a.	喷墨的 (2A)
inkjet printer	喷墨打印机 (2A)
innocuous /'ɪnɒkjʊəs/ a.	无害的; 不得罪 (或冒犯) 人的 (1C)
innovation /,ɪnəʊ'veɪʃən/ n.	革新, 创新 (7A)
input stream	输入 (信息) 流 (1A)
instant messaging	即时通信, 即时消息 (7A)
instantaneous /,ɪnstən'teɪniəs/ a.	瞬间的, 即刻的 (3B)
instruction set	指令集 (1A)
instrumental /,ɪnstru'mentəl, ɪnstre-/ a.	起作用的, 有帮助的 (<i>in, to</i>) (8C)
insulin /'ɪnsjʊlɪn; 'ɪnsə-/ n.	胰岛素 (制剂) (9C)
insulin pump	胰岛素泵 (9C)
intact /ɪn'tækt/ a.	完整无缺的; 未受损伤的 (9B)
integer /'ɪntɪdʒə/ n.	整数 (5B)
integral /'ɪntɪgrəl/ a.	构成整体所必需的, 基本的 (1B)
integrated /'ɪntɪgreɪtɪd/ a.	集成的, 综合的, 一体化的 (1A)
integrated circuit	集成电路 (1A)
integrated user interface	综合用户接口 (2C)
integrator /'ɪntɪgreɪtə/ n.	积分器 (1A)
intellectual property	知识产权 (12A)
intent /ɪn'tent/ n.	意图, 目的 (5C)
interactive /,ɪntər'æktɪv/ a.	交互 (式) 的 (6C)
intercept /,ɪntə'sept/ v.	拦截; 截取 (12C)
interchange /,ɪntə'tʃeɪndʒ/ v.	交换, 互换 (2C)
interchangeable /,ɪntə'tʃeɪndʒəbəl/ a.	可交换的, 可互换的 (7C)
interdisciplinary /,ɪntə'dɪsplɪnəri/ a.	学科间的, 跨学科的 (1B)
interface /'ɪntəfeɪs/ n.	界面; 接口 (1B)
interfacing /'ɪntəfeɪsɪŋ/ n.	接口技术 (1B)
interleave /'ɪntəli:v/ v.	交错, 交叉, 交替; 隔行扫描 (5A)
intermediary /,ɪntə'mɪ:diəri/ n.	媒介 (物); 调解人; 中间人 (6A)
intermediate language	中间语言, 中级语言 (3A)
intermix /,ɪntə'mɪks/ v.	(使) 混合 (<i>with</i>) (5B)
internet /'ɪntənɪt/ n.	互联网, 互连网 (7A)
Internet service provider	因特网服务提供商 (7B)
internetwork /,ɪntə'netwɜ:k/ n.	互联网, 互连网 (7C)
interoperability /'ɪntər,ɒpə'reɪbɪlɪti/ n.	互操作性, 互用性 (4C)
interoperable /,ɪntər'ɒpərəbl/ a.	可互操作的, 能互用的 (9A)
interpreted code	解释执行的代码 (3B)
interpreted language	解释执行的语言 (4A)
interpreter /ɪn'tə:prɪtə/ n.	解释程序, 解释器 (3B)
interpreter program	解释程序 (4A)
interprocess communication	进程间通信 (7A)
interrupt /,ɪntə'rapt/ n.	中断 (2B)
interrupt handler	中断处理程序 (2B)
interrupt signal	中断信号 (2B)
intersection /,ɪntə'sekʃən/ n.	交, 相交, 交集; 交点 (6A)
intertwine /,ɪntə'twɪn/ v.	(使) 缠结, (使) 缠绕在一起 (3B)
interweave /,ɪntə'wi:v/ v.	(使) 交织 (6B)
intranet /'ɪntrənɪt/ n.	内部网, 企业内部网, 内联网, 内网 (4C)
intrinsically /ɪn'trɪnsɪkəli/ ad.	固有地; 本质上; 内在地 (4C)
intrusion /ɪn'tru:ʒən/ n.	侵入; 打扰 (6C)

inundate /'inəndeɪt/ v.	淹没; (似洪水般) 布满 (8C)
inventory /'inventəri; -tɔ:ri/ n. & v.	存货 (清单), 库存 (9C) / 为...开列存货清单; 编制...的目录; 盘存 (9B)
inverted /in've:tɪd/ a.	反向的; 倒置的 (6A)
investor /in'vestə/ n.	投资者 (12A)
invocation /,ɪnvə'keɪʃən/ n.	调用 (4C)
invoke /in'vəuk/ v.	调用; 激活 (10B)
involved /in'vɒlvd/ a.	(由于复杂或混乱而) 难处理的, 棘手的; 复杂的 (6A)
irony /'aɪərəni/ n.	反语; 讽刺文体 (11A)
irrespective /,ɪrɪ'spektɪv/ a.	不考虑的; 不顾的 (of) (5A)
iteration /,ɪte'reɪʃən/ n.	迭代 (法), 重复 (5A)
iterative /'ɪtərətɪv, -reɪt-/ a.	重复的; 迭代的 (6C)

J

jack /dʒæk/ n.	插座; 插口 (9C)
jewelry /'dʒu:əlri/ n.	〈美〉[总称] 珠宝, 首饰 (8C)
joystick /'dʒɔɪstɪk/ n.	控制杆, 操纵杆, 游戏杆 (1C)
junk /dʒʌŋk/ n.	废旧物品; 破烂, 垃圾 (11C)

K

kernel /'ke:nəl/ n.	内核, 内核程序, 核心程序 (2B)
keypad /'ki:pæd/ n.	小键盘, 袖珍键盘; 键区 (1C)
keyword /'ki:wə:d/ n.	关键词, 关键字 (8C)
knowledge base	知识库 (6C)

L

laptop /'læptɒp/ n. & a.	膝上型计算机 (1A) / 膝上型的 (2A)
laser printer	激光打印机 (2A)
launching pad	发射台; 跳板; 起点 (10C)
layered /'leɪəd/ a.	分层的 (2C)
legality /li'gælɪti/ n.	合法 (性) (7A)
legitimate /li'dʒɪtɪmɪt/ a.	合法的; 正当的 (10A)
lever /'li:və; 'le-/ n.	(杠) 杆; 控制杆 (2A)
leverage /'li:vəɪdʒ; 'le-/ v.	充分利用 (9A)
library /'laɪbrəri/ n.	库, 程序 (或文件、对象) 库 (2C)
library routine	库程序, 库存 (子) 程序, 程序库例行程序 (4A)
licensing /'laɪsənsɪŋ/ n.	发给许可证; 许可, 特许 (9A)
life cycle	生命周期, 生存周期, 寿命周期 (1B)
light pen	光笔 (2A)
likelihood /'laɪklihud/ n.	可能, 可能性 (10A)
limelight /'laɪmlaɪt/ n.	公众注意的中心 (9B)
linear /'liːniə/ a.	线 (性) 的; 直线的 (3C)
linguistic /lɪŋ'gwɪstɪk/ a.	语言 (学) 的 (3A)

linguistics /lin'gwistiks/ n.
 linker /'lɪŋkə/ n.
 liquid crystal display
 list box
 listing /'listɪŋ/ n.
 listserv /'listɜ:v/ n.
 live-and-let-live
 local area network
 localize /'ləukəlaɪz/ v.
 locator /ləu'keɪtə; 'ləukeɪtə/ n.
 locator software
 lock-in /'lɒkɪn/ n.
 log /'lɒg/ n. & v.

logic bomb
 logic language
 logical element
 login /'lɒɡɪn; 'lɔ:g-/ n.
 logistics /ləu'dʒɪstɪks/ n.
 logo /'ləʊɡəʊ/ n.
 loom /lu:m/ n.
 looping /'lu:pɪŋ/ n.
 loosely coupled system
 lowercase /'ləʊə'keɪs/ a.
 low-tech /ləu'tek/ a.
 lurk /lə:k/ v.

语言学 (1B)
 连接程序, 链接程序 (3B)
 液晶显示 (器) (2A)
 列表框 (12B)
 列表; 一览表; 目录 (12B)
 邮件发送清单 (或邮件列表) 管理程序 (11A)
 自己活也让别人活; 互相宽容; 互不相扰 (11C)
 局域网 (6C)
 使局部化, 使本地化 (2C)
 定位器, 定位符 (8C)
 定位软件 (12C)
 锁定; 同步 (9A)
 (运行) 记录; (系统) 日志 (6B) / 登录, 注册, 进入系统 (*on, in*) (10A)
 逻辑炸弹 (10C)
 逻辑语言 (3A)
 逻辑元件 (1A)
 注册, 登录, 进入系统 (8B)
 后勤 (学) (9B)
 标识, 标志图, 徽标 (11B)
 织机 (1A)
 循环; 构成环形 (3C)
 松散耦合系统 (4C)
 (字母) 小写的 (8C)
 低技术的 (11B)
 潜伏; 暗藏; 潜行 (6B)

M

machine vision
 macro /'mækroʊ/ n.
 magnetic disk
 magneto-optical /mæɡ.ni:təu'optɪkəl/ a.
 magnify /'mæɡnɪfaɪ/ v.
 mailing list
 main memory
 main page
 mainframe /'meɪnfreɪm/ n.
 mainstream /'meɪnstri:m/ a.
 maintenance hook
 malicious /mə'lɪʃəs/ a.
 mall /mɔ:l, məl/ n.
 malware /'mælwɛə/ n.
 manager /'mænɪdʒə/ n.
 manifest /'mænɪfest/ v.
 map /mæp/ v.
 marketer /'mɑ:kɪtə/ n.

机器视觉 (9B)
 宏, 宏指令 (3A)
 磁盘 (2B)
 磁光的 (2A)
 放大, 扩大 (10B)
 邮件发送清单, 邮件列表 (11A)
 主存 (存储器) (2B)
 主页 (8C)
 主机, 大型机 (1A)
 主流的 (5C)
 维护陷阱 (10C)
 恶意的 (3B)
 (非露天的) 购物区, 购物街 (9B)
 恶意软件 (10B)
 管理程序, 管理器 (2B)
 显示; 使显现 (5B)
 映射; 变换, 变址 (2C) / 详细安排, 筹划 (*out*) (11B)
 在市场上做买卖的人; 专营特定商品的商人 (或商号) (1C)

marketplace /'mɑ:kɪtpleɪs/ n.	市场, 集市, 市集 (12A)
masquerade /,mæskə'reɪd, mɑ:s-/ n. & v.	假面舞会; 伪装; 假扮 (10A) / 假冒; 假扮 (10B)
mass storage	海量存储器, 大容量存储器 (2B)
mass-produce /'mæsprə'dju:s/ v.	(使用机器) 大量生产 (1B)
matrix /'meɪtrɪks/ n.	([复] -trices /-trɪsɪz/ 或 -trixes) (矩) 阵 (2A)
maximize /'mæksɪmaɪz/ v.	使增加 (或扩大) 到最大限度 (4B)
measured /'meɪʒəd/ a.	慢而稳的; 慎重的, 几经斟酌的 (12A)
mediate /'mi:dieɪt/ v.	调解; 传递 (4C)
medic /'medɪk/ n.	〈口〉(随军急救) 卫生队队员; 医科学生; 医生 (8C)
medication /,medɪ'keɪʃən/ n.	药物治疗; 药物 (8C)
medicinal /me'dɪsɪnəl/ a.	(医) 药的; 药用的; 有疗效的 (8C)
memo /'meməu, 'mi:-/ n.	〈口〉备忘录 (= memorandum) (11A)
memory /'meməri/ n.	存储器, 内存 (1A)
memory location	存储单元 (4A)
memory-resident /'meməri'rezɪdənt/ a.	内存驻留的, 常驻内存的 (10B)
merchandise /'mɜ:tʃəndaɪz, -daɪs/ n.	[总称] 商品 (9B)
merge /mɜ:dʒ/ v.	合并; 结合 (3A)
mesh /meʃ/ n.	网; 网格; 网状结构; 网状网络 (7B)
message board	留言板, 消息板 (11C)
messaging /'mesɪdʒɪŋ/ n.	消息接发, 通信 (1C)
metamorphic /,metə'mɔ:fɪk/ a.	变形的 (10B)
metaphor /'metəfə/ n.	隐喻; 比喻 (12C)
methodology /,meθə'dɒlədʒi/ n.	(学科的) 一套方法; 方法论 (4B)
metropolitan /,metrə'pɒlɪtən/ a.	大城市的, 大都会的 (7A)
metropolitan area network	城域网 (7A)
microchip /'maɪkrəʊtʃɪp/ n.	微芯片 (1A)
microcomputer /'maɪkrəʊkəm'pjʊ:tə/ n.	微型计算机 (1A)
microcontrol /'maɪkrəʊkən'trəʊl/ v.	微控制 (1C)
microcontroller /'maɪkrəʊkən'trəʊlə/ n.	微控制器 (1C)
microminiaturization /'maɪkrəu,mɪnɪtʃəraɪ'zeɪʃən/ n.	微小型化, 超小型化 (1A)
microprocessor /,maɪkrəʊ'prəʊsesə/ n.	微处理器 (1A)
microwave oven	微波炉 (1A)
middleware /'mɪdlweə/ n.	中 (间) 件, 中间设备 (12B)
milieu /'mi:lje:, mi:'lje:/ n.	(周围) 环境 (6C)
millennium /mɪ'lenɪəm/ n.	一千年; 千禧年 (11C)
millisecond /'mɪlɪsekənd/ n.	毫秒 (2B)
mimic /'mɪmɪk/ v.	模仿 (1A)
miniature /'mɪnɪtʃə: 'mɪnɪetʃə/ a.	小型的, 微小的 (11B)
miniaturization /,mɪnɪtʃəraɪ'zeɪʃən; mɪnɪetʃərə-/ n.	小型化, 微型化 (9B)
minicomputer /,mɪnɪkəm'pjʊ:tə/ n.	小型计算机 (1A)
minimal /'mɪnɪməl/ a.	最小的; 最低限度的 (2B)
minimize /'mɪnɪmaɪz/ v.	使减少到最低限度 (3A)
minuscule /'mɪnəskju:l, mɪ'nəs-/ a.	非常小的, 微小的 (9B)
mischievous /'mɪstʃɪf/ n.	恶作剧; 淘气; 损害 (11C)
misnomer /,mɪs'neɪmə/ n.	错误 (或使用不当) 的名字 (或名称) (6C)
mnemonic /ni:'mɒnɪk/ a.	助记的; 记忆的 (8A)
mnemonic address	助记地址 (8A)
mnemonic name	助记名, 缩写名 (8A)
mobile phone	移动电话 (9C)
mobile telephone	移动电话 (12A)

modeling /'mɒdəlɪŋ/ n.	建模, 模型化 (4B)
modification /ˌmɒdɪfɪ'keɪʃən/ n.	修改, 更改 (5A)
modular /'mɒdjʊlə/ a.	模块化的 (11B)
modulate /'mɒdjuleɪt/ v.	调制 (9B)
modulator /'mɒdjuleɪtə/ n.	调制器 (2A)
module /'mɒdju:l/ n.	模块 (2A)
molecular /məʊ'lekjələ/ a.	分子的 (1C)
mom /mɒm/ n.	〈主美口〉妈妈 (3B)
monetary /'mɒnɪtəri/ a.	货币的; 金钱的 (10A)
monopoly /mə'nɒpəli/ n.	垄断; 独有 (11C)
mortar /'mɔ:tə/ n.	砂浆, 灰浆 (12C)
motherboard /'mʌðəbɔ:d/ n.	母板, 主板 (2A)
motion picture	〈美〉影片, 电影 (7A)
motivation /ˌməʊtɪ'veɪʃən/ n.	动机, 诱因 (10C)
multicast /'mʌltɪkɔ:st/ n.	组播 (通信), 多址通信 (4C)
multicast protocol	多址通信协议 (4C)
multidimensional /ˌmʌltɪ'dɪmənʃənəl, -daɪ'm-/ a.	多维的 (6C)
multidimensional array	多维数组 (3C)
multimedia /ˌmʌltɪ'mɪ:diə/ a.	〈使用〉多媒体的 (4C)
multipartite /ˌmʌltɪ'pɑ:tɪt/ a.	分成多部分的 (10C)
multitasking /'mʌltɪ,tɑ:skɪŋ/ n.	多任务 (处理) (2B)
multitenant /ˌmʌltɪ'tenənt/ a.	多租户的 (9A)
multithreaded /'mʌltɪ,θredɪd/ a.	多线程的 (3B)
multithreading /'mʌltɪ,θredɪŋ/ n.	多线程操作 (3B)
mundane /mʌn'deɪn/ a.	世俗的; 平凡的 (9C)

N

name server	名字服务器 (8A)
native code	本机 (代) 码 (3B)
navigate /'neɪvɪgeɪt/ v.	航行 (于); (为...) 领航; 指引 (2A)
navigation /ˌnævɪ'geɪʃən/ n.	导航; 航行 (9B)
navigation bar	导航条 (11B)
navigation page	导航页面 (11B)
navigator /'nævɪgeɪtə/ n.	导航仪, 导航系统; 航海者 (9B)
negligible /'neglɪdʒəbl/ a.	可略而不计的 (4C)
netiquette /'netɪket/ n.	网络礼节, 网规 (<i>network etiquette</i> 的缩合) (11A)
network /'netwɜ:k/ v.	联网 (2C)
network identifier	网络标识符 (8A)
networking /'net.wɜ:kɪŋ/ n.	连网, 联网, 建网 (1A)
neural /'njuərəl/ a.	神经的 (1A)
neural network	神经网络 (1A)
neuron /'njuərən/ n.	神经元, 神经细胞 (1B)
neurophysiology /'njuərəʊ,fɪzɪ'ɒlədʒi/ n.	神经生理学 (1B)
neutrality /nju:'træləlɪti/ n.	中立 (11C)
neutralize /'nju:trəlaɪz; 'nu:-/ v.	使成为无效; 抵消 (10B)
niche /nɪʃ, ni:f/ n.	缝隙市场, “利基” 市场 (1C)
node /nəʊd/ n.	节点 (4C)
node page	节点页面 (11B)

nondeterministic /ˌnɒndɪtɜːmɪ'nɪstɪk/ a.	不确定的, 非确定性的 (10B)
nonetheless /ˌnɒnðə'les/ ad.	尽管如此, 然而 (1C)
nonprofit /ˌnɒn'prɒfɪt/ a.	非营利的 (8A)
nonrepudiation /ˌnɒnrɪˌpjuːdi'eɪʃən/ n.	不拒绝; 不否认; 不可抵赖 (10A)
nonvolatile memory	非易失性存储器 (2A)
normalize /ˈnɔːməlaɪz/ v.	使正常 (或标准、规范、正规、规格) 化 (6A)
notation /ˌnəʊ'teɪʃən/ n.	标记法; 记号 (3C)
notebook /ˈnəʊtbʊk/ n.	笔记本 (式计算机) (9C)
notebook computer	笔记本 (9C)
notification /ˌnəʊtɪfɪ'keɪʃən/ n.	通知 (4C)
novelist /ˈnɒvəlɪst/ n.	(长篇) 小说家 (5C)
novice /ˈnɒvɪs/ n.	新手, 初学者 (5C)
nugget /ˈnʌɡɪt/ n.	小块; 天然金块; 有价值的东西 (6C)
numeric(al) /ˌnʊː'merɪk(əl)/ a.	数字的; 数值的 (1A)
numerical analysis	数值分析 (1B)

O

object code	目标 (代) 码 (4A)
object diagram	对象图 (4B)
object-oriented language	面向对象语言 (3A)
obsolete /ˈɒbsəli:t, ˌɒbsə'li:t/ a.	废弃的; 淘汰的; 过时的 (6A)
obstruct /əb'strʌkt, ɒb-/ v.	阻塞; 阻碍 (8B)
offering /ˈɒfərɪŋ; ˈɔː-/ n.	待售品; 提供之物; 捐献物 (12B)
office tower	办公大楼 (9B)
offset /ˈɒfset; ˈɔːf-/ v.	(出于比较等目的) 把...并列, 衬托出; 补偿; 抵消 (11B)
offshoot /ˈɒfʃu:t/ n.	支族, 旁系; 衍生事物 (3B)
off-the-shelf /ˈɒfðə'self/ a.	现成的, 非专门设计 (或定制) 的 (5A)
oligomorphic /ˌɒlɪɡəʊ'mɔːfɪk/ a.	寡形的 (10B)
omission /əʊ'mɪʃən/ n.	省略; 遗漏; 疏忽 (4B)
on-board /ˈɒn'bɔːd, ˈɔːn-/ a.	车载的; 机载的; 舰载的 (9C)
one-dimensional array	一维数组 (3C)
ongoing /ˈɒŋɡeɪɪŋ/ a.	正在进行的, 不间断的 (1A)
operator /ˈɒpəreɪtə/ n.	(运) 算子, 算符 (8C)
optimal /ˈɒptɪmə/ a.	最适的, 最优的, 最佳的 (1B)
optimization /ˌɒptɪmaɪ'zeɪʃən/ n.	最佳化, (最) 优化 (1B)
optimize /ˈɒptɪmaɪz/ v.	使优化, 使最佳化 (2C)
ordinal /ˈɔːdlɪnəl/ a.	顺序的 (3C)
ordinal number	序数 (3C)
orient /ˈɔːrɪent/ v.	给...定向; 使适应; 使面向 (3A)
orientation /ˌɔːrɪen'teɪʃən/ n.	方向; 定向 (11A)
originate /ə'rɪdʒɪneɪt/ v.	发源; 产生 (1C)
originator /ə'rɪdʒɪneɪtə/ n.	创始人; 起源 (10A)
outbreak /ˈaʊtbreɪk/ n.	爆发 (10B)
outgoing /ˈaʊtɡeɪɪŋ/ a.	往外去的, 离去的 (11C)
outgrowth /ˈaʊtɡreʊθ/ n.	发展结果; 产物 (3A)
outlast /ˌaʊt'lɑːst; -læst/ v.	比...持久 (4C)
outlet /ˈaʊtlet, -lɪt/ n.	销路; 商店 (12C)

outlier /'aut.laɪə/ n.
 outright /'autrait/ a.
 outsource /aut'sɔ:s/ v.
 outweigh /,aut'wei/ v.
 overhead /'əuvəhed/ n.
 overlap /,əuve'læp/ v.
 overload /,əuve'ləud/ v.
 oversee /,əuve'si:/ v.
 overshadow /,əuve'ʃædəu/ v.
 oversight /'əuvesait/ n.
 overturn /,əuve'tə:n/ v.
 overview /'əuvevju:/ n.
 overwrite /,əuve'rait/ v.

处于外面的东西; 外面部分 (6C)
 完全的, 彻底的 (4C)
 (将...) 外包 (9A)
 在价值 (或重要性、影响等) 方面超过 (3B)
 经常 (或管理、间接) 费用 (3B)
 重叠, 层叠 (1B)
 使过载, 使超载 (10A)
 监视; 检查; 管理 (8A)
 使相形见绌 (1A)
 监督; 照管 (12C)
 推翻; 废除 (1A)
 概述; 概观 (1A)
 盖写, 改写, 重写 (2A)

P

pacemaker /'peɪs.meɪkə/ n.
 package /'pækɪdʒ/ n.
 paging /'peɪdʒɪŋ/ n.
 palmtop /'pɑ:mtop/ n.
 paper-tape reader
 paradigm /'pærədəɪm/ n.
 parameter /pe'ræməti:/ n.
 paraphrase /'pærəfreɪz/ v.
 parasitic /,pærə'sitik/ a.
 parking lot
 parking meter
 participant /pɑ:'tɪsɪpənt/ n.
 partition /pɑ:'tɪʃən/ v. & n.
 patchwork /'pætʃwɜ:k/ n.
 patent /'peɪtənt/ n.
 path expression
 pay-as-you-go /'peɪəzju'gəu/ n. & a.
 paycheck /'peɪtʃɛk/ n.
 payload /,peɪləud/ n.
 payroll /'peɪrɒl/ n.
 peer /piə/ n.
 peer-to-peer /'pi:tə'piə/ a.
 per se /,pə:'seɪ, -'si:/
 perforated /'pə:fəreɪtɪd/ a.
 periodic /,piəri'ɒdɪk/ a.
 peripheral /pe'rɪfərəl/ a. & n.
 peripheral device
 perishable /'perɪʃəbəl/ a.
 persistence /pə'sɪstəns/ n.
 persistent /pə'sɪstənt/ a.
 personal digital assistant
 personal organizer

起搏器 (9C)
 程序包, 软件包 (3B)
 页面调度 (技术或方法), 分页 (2B)
 掌上型计算机 (1A)
 纸带阅读器 (1A)
 范例; 范式 (4B)
 参数, 参量 (3A)
 将...释义 (或意译) (11A)
 寄生的 (10C)
 〈美〉停车场, 停车区 (9C)
 汽车停放计时器, 汽车停放收费计 (9C)
 参加者, 参与者 (9B)
 划分, 分区, 分割 (5A) / 分割; 划分;
 (分割而成的) 部分 (5B)
 拼缀物; 拼凑的东西, 杂烩 (9C)
 专利 (权) (1A)
 路径表达式 (2B)
 即付即用 (的) (9A)
 〈美〉支付薪金 (或工资的) 支票 (6A)
 有效载荷 (10C)
 工资表; 在职人员名单; 工薪总额 (6A)
 同级设备; 同级, 同层, 对等层 (4C)
 对等的 (7A)
 〈拉〉自身, 本身 (5B)
 穿孔的 (1A)
 周期 (性) 的; 定期的 (10B)
 外围的, 外部的 (2B) / 外围设备, 外部设备 (4C)
 外围设备, 外部设备 (2B)
 易腐败的, 易腐烂的 (9B)
 持续性, 持久性 (4C)
 持续的, 持久的 (4C)
 个人数字助理 (12A)
 备忘录本 (1C)

personalize /'pɜ:sənəlaiz/ v.	使个性化 (9A)
pertinent /'pɜ:tinənt/ a.	相关的 (5C)
pervasive /pə'veisiv/ a.	遍布的; 普遍的 (4C)
petabyte /'petəbaɪt/ n.	拍字节, 千万亿字节 (6C)
pharmaceutic(al) /fɑ:mə'sju:tik(əl)/ a.	制药的; 药(物)的; 药用的 (6A)
phishing /'fɪʃɪŋ/ n.	网络钓鱼 (10B)
phosphorescent /fɒsfə'resənt/ a.	发磷光的 (2A)
photocopier /fəʊtəu'kɒpiə/ n.	复印机 (2A)
physiologist /fizi'ɒlədʒɪst/ n.	生理学家 (1B)
pickup /'pɪkʌp/ n.	搭车; 接人; 提货 (9C)
pictorial /pɪk'tɔ:riəl/ a.	用图表示的, 图示的; 形象化的 (2B)
pinpoint /'pɪnpɔɪnt/ v.	精确地确定...的位置; 确定 (5B)
piracy /'paɪərəsi/ n.	侵犯版权, 盗版, 剽窃 (10A)
pirate /'paɪərət/ v.	剽窃; 盗用; 非法翻印 (11C)
pit /pɪt/ n.	凹区, 凹槽 (2A)
player piano	自动钢琴 (4A)
playwright /'pleɪraɪt/ n.	剧作家 (5C)
plaza /'plɑ:zə, 'plæ-/ n.	(城市建筑群中间或附近的) 购物区; (城市中的) 广场 (8A)
pluck /plʌk/ v.	拔; 摘, 采 (11C)
plus /plʌs/ ad.	〈口〉外加地; 另外
pointer /'pɔɪntə/ n.	指针 (光标); 指示字, 指示符 (2A)
police /pə'li:s/ v.	维持...的治安; 管理; 监督 (11C)
pollutant /pə'lu:tənt/ n.	污染性物质 (9B)
polymorphic /pəli'mɔ:fɪk/ a.	多(种)形(式)的, 多态的 (10B)
polymorphism /pəli'mɔ:fɪzəm/ n.	多态性, 多形性 (3B)
pool /pu:l/ v.	把...集中一起用, 共用 (8A)
pornography /pɔ:'nɒgrəfi/ n.	[总称] 色情(或淫秽)作品 (11C)
port /pɔ:t/ n.	端口, 通信口 (8B)
port number	端口号 (8B)
portability /pɔ:te'bɪlɪti/ n.	可移植性; 便于携带 (5C)
portal /'pɔ:təl/ n.	门户, 入口 (12A)
postal money order	邮政汇票 (12C)
pothole /'pəʊhəl/ n.	路面凹坑, 坑洞 (9C)
power grid	电网 (9B)
pragmatic /præg'mætɪk/ a.	讲究实际的, 重实效的; 实用主义的 (5C)
precede /,pri:'si:d, pri-/ v.	处在...之前, 先于 (4A)
precursor /,pri:'kɜ:sə, pri-/ n.	先驱 (1A)
predate /pri:'deɪt/ v.	在日期上早于(或先于) (10C)
predecessor /'pri:dɪsə'sə, 'pre-/ n.	前任; 前身; 前驱, 先驱 (2C)
preempt /,pri:'empt, pri-/ v.	预先制止; 抢先, 先占, 预占 (2B)
prehistoric /,pri:'hi:stɒrɪk/ a.	史前的 (2B)
premature /,preme'tʃue, pri:-/ a.	不成熟的, 仓促的, 过早的 (5A)
premium /'pri:miəm/ n.	奖品; 额外补贴; 加付款 (4B)
preparatory /pri'pære'tɔ:ri, -tɔ:ri/ a.	准备性的, 预备性的 (5B)
preprocessor /pri:'preʊsesə/ n.	预处理程序, 预处理器 (3B)
presentation software	演示软件 (11A)
print server	打印服务器 (7A)
printer /'prɪntə/ n.	打印机 (1A)
printing press	印刷机 (11C)

privacy /'privesi, 'prai-, 'prai-/ n.
 privacy policy
 privileged /'privilidʒd/ a.
 procedural language
 procedure call
 process switch
 process table
 procure /preu'kjue/ v.
 profitable /'prɒfitebl/ a.
 program module
 program unit
 programmability /,preugræme'biliti/ n.
 programming construct
 proliferation /preu.li'fe'reiʃən/ n.
 prompt /prompt/ v.
 prone /preun/ a.
 proofread /'pru:fri:d/ v.
 proofreading /'pru:fri:diŋ/ n.
 propagate /'prɒpegelt/ v.
 propagation /,prɒpe'geiʃən/ n.
 proprietary /preu'praɪetəri/ a.
 protocol /'prəutekəl/ n.
 protocol suite
 prototype /'prəutetaip/ n.
 prototyping /'prəute'taɪpɪŋ/ n.
 prover /'pru:və/ n.
 provision /preu'vɪʒən/ v.
 punch card
 punched card
 punctuation /,pʌŋktju'eɪʃən/ n.
 punctuation mark
 pursuit /pe'sju:t/ n.

隐私; 秘密 (1A)
 隐私政策 (12C)
 享有特权的; 特许的 (6B)
 过程语言 (3A)
 过程调用 (2C)
 进程转换 (2B)
 进程表 (2B)
 取得, 获得; 采办 (5A)
 有利可图的, 有盈利的 (9C)
 程序模块 (3A)
 程序单元 (5A)
 可编程 (序) 性 (1C)
 编程结构 (3C)
 激增; 扩散 (10A)
 提示, 提醒 (4A)
 有...倾向的, 易于...的 (12B)
 校对 (11B)
 校对 (11B)
 传播; 繁殖; 蔓延 (10C)
 传播; 繁殖; 蔓延 (10C)
 专有的, 专用的; 专利的 (7A)
 协议 (2C)
 协议组, 协议集 (7A)
 原型; 样机 (1A)
 原型法; 原型开发 (或设计); 样机研究 (5A)
 证明程序 (3B)
 供应, 提供 (9A)
 穿孔卡片 (4A)
 穿孔卡片 (1A)
 标点符号 (3A)
 标点符号 (8C)
 追求; 追赶 (6C)

Q

quality seal
 quarantine /'kwɒrənti:n/ v.
 query /'kwɪəri/ n.
 quip /kwɪp/ n.
 quotation marks
 qwerty /'kwe:ti/ n.

质量图章 (12C)
 隔离 (10B)
 查询 (2C)
 妙语, 俏皮话 (11A)
 [复] (一对) 引号 (8C)
 〈口〉标准打字机键盘 (亦作 qwerty keyboard) (1C)

R

radio button
 radio frequency
 radio-frequency identification
 rage /reɪdʒ/ v.

单选 (按) 钮, 小圆按钮 (4B)
 射频; 无线电频率 (9B)
 射频识别 (9B)
 发怒; 激烈进行 (11C)

railroad car	〈美〉火车车厢 (8C)
rampant /'ræmpənt/ a.	猖獗的; 无约束的 (12A)
random /'rændəm/ a.	随机的; 任意的 (2A)
random-access memory	随机 (存取) 存储器 (2A)
read-only memory	只读存储器 (2A)
real time	实时 (6C)
realign /,ri:'ə'lain/ v.	重新调整 (4A)
recipe /'resipi/ n.	烹饪法, 食谱; 处方 (8C)
recipient /ri'sipiənt/ n.	接受者; 收件人 (4C)
recourse /ri'kɔ:s; 'ri:kɔ:rs/ n.	追索权; 偿还请求; 求助 (10B)
recur /ri'ke:/ v.	再发生; 反复出现 (5C)
redundancy /ri'dʌndənsi/ n.	冗余 (2C)
redundant /ri'dʌdənt/ a.	冗余的 (7B)
reference /'refərəns/ n. & v.	引用, 参考; 基准 (3C) / 引用, 参考 (2B)
reformat /,ri:'fɔ:mæt/ v.	重新格式化 (4A)
refrigeration /re'fridʒə'reiʃən/ n.	冷藏; 制冷 (9B)
refund /'ri:fʌnd/ n.	退款; 退还 (11C)
register /'redʒistə/ n.	寄存器 (3A)
registrar /,redʒi'stra:, 'redʒis-/ n.	域名注册服务商 (8A)
registration /,redʒi'streiʃən/ n.	登记; 注册 (6B)
registry /'redʒistri/ n.	注册表; 登记簿 (10C)
regression /ri'greʃən/ n.	回归 (5B)
regression test	回归测试 (5B)
relational database management system	关系数据库管理系统 (6A)
relational language	关系语言, 相关语言 (3A)
relay /ri:'lei/ v.	接力传送; 传递; 转播 (1C)
release /ri:'li:s/ n.	(程序或软件的) 版本; 发布 (3B)
reliance /ri'laɪəns/ n.	依靠, 依赖 (4C)
remote sensing	遥感 (6C)
rental /'rentəl/ n.	租赁, 出租 (9C)
repeater /ri'pi:tə/ n.	中继器, 转发器 (7C)
repetitive /ri'petitiv/ a.	重复的 (1B)
replica /'replika/ n.	复制品, 拷贝 (6A)
replicate /'replikeit/ v.	复制, 重复 (6A)
repository /ri'pɒzɪtəri/ n.	存储库, 仓库 (2C)
reprisal /ri'praɪzəl/ n.	报复 (11C)
repudiation /ri,pju:di'eɪʃən/ n.	拒绝; 否认 (12C)
reputable /'repjutable/ a.	声誉好的 (12C)
reseller /,ri:'selə/ n.	转售商, 代销商 (12C)
reside /ri'zaid/ v.	驻留, 常住 (2B)
residential /,rezi'denʃəl/ a.	居住的; 住宅的 (9B)
resiliency /ri'ziliənsi/ n.	弹性; 复原力 (9A)
resize /ri:'saɪz/ v.	调整大小, 重定尺寸 (4B)
respondent /ri'spɒndənt/ n.	〈主美〉(民意测验等的) 调查对象, (调查表、问题表的) 答卷人 (10A)
retailer /'ri:teɪlə/ n.	零售商; 零售店 (9B)
retailing /'ri:teɪlɪŋ/ n.	零售业 (9B)
retina /'retinə/ n.	视网膜 (12A)
retina scanner	视网膜扫描仪 (12A)
retrieval /ri'tri:vəl/ n.	检索 (2A)

retrieve /ri'tri:v/ v.
 revoke /ri'veuk/ v.
 revolutionize /,revə'ljʊ:fənaiz/ v.
 rigid /'ridʒid/ a.
 rigidity /ri'dʒiditi/ n.
 rigorous /'riggerəs/ a.
 ring topology
 roam /rəʊm/ v.
 roboticist /rəu'bɒtɪsɪst/ n.
 robotics /rəu'bɒtɪks/ n.
 robust /rəu'bʌst, 'rəubʌst/ a.
 robustness /rəu'bʌstnis, 'rəubʌst-/ n.
 rodent /'rəʊdənt/ n.
 rollback /'rəʊlbæk/ n.
 romance /rəʊ'mæns, 'rəumæns/ n.
 rootkit /'ru:tkit/ n.
 roulette /ru:'let/ n.
 roulette wheel
 route /ru:t/ v.
 router /'ru:tə/ n.
 routine /ru:'ti:n/ n.
 routing /'ru:tiŋ/ n.
 routing algorithm
 routing path
 runtime /'rʌntaɪm/ n.
 runway /'rʌnwei/ n.

检索 (1B)
 撤销; 废除 (6A)
 使发生革命性剧变 (9C)
 硬的 (2A)
 严格; 刻板 (4B)
 严密的; 严格的 (3A)
 环型拓扑结构 (7A)
 漫步; 漫游 (9C)
 机器人 (制造操作) 专家 (1B)
 机器人学; 机器人技术 (1B)
 健壮的, 坚固的 (1B)
 健壮性, 坚固性 (6C)
 啮齿目动物 (如鼠、松鼠、河狸等) (9C)
 回滚, 回退; 重新运行, 重算 (2C)
 浪漫文学, 传奇文学 (5C)
 根工具包 (10B)
 轮盘赌 (11C)
 (轮盘赌台上的) 轮盘 (11C)
 (按特定路径) 发送, 传递 (7C)
 路由器 (7B)
 例程, 例行程序 (4A)
 路由选择 (4C)
 路由算法 (4C)
 路由选择通路 (7B)
 运行时刻 (3B)
 (机场的) 跑道 (7C)

S

safeguard /'seɪfɡɑ:d/ v.
 sandbox /'sændbɒks/ n.
 sarcasm /'sɑ:kæzəm/ n.
 scalability /,skeɪlə'bɪlɪti/ n.
 scalable /'skeɪləbl/ a.
 scale /skeɪl/ v.
 scaled-down /'skeɪld'daʊn/ a.
 scanner /'skæne/ n.
 scenario /si'nɑ:riəʊ/ n.
 scheduler /'ʃedʒu:əle; 'skedʒu:əler/ n.
 schema /'ski:mə/ n.
 scroll /skrəʊl/ v. & n.
 scroll bar
 seamless /'si:mlis/ a.
 search engine
 secrecy /'si:kresi/ n.
 Secure Sockets Layer
 segment /'segment/ n.
 segmentation /,segmen'teɪʃən/ n.
 segregate /'segrɪgeɪt/ v.

保护, 维护 (9A)
 (供儿童在其中做堆沙游戏的) 沙池 (10B)
 讽刺, 挖苦 (11A)
 可缩放性, 可伸缩性 (6C)
 可缩放的, 可伸缩的 (6C)
 攀登; (阶梯) 逐步升高 (9A)
 按比例缩减的 (1C)
 扫描仪; 扫描程序 (2A)
 方案; 情况; 脚本 (9A)
 调度程序 (2B)
 模式; 图表; 纲要 (2C)
 滚动 (4B)
 滚动条 (4B)
 无缝的 (6C)
 搜索引擎 (6C)
 秘密; 保密 (10A)
 安全套接层 (协议) (12C)
 段; 片段; 程序段 (3A)
 (程序) 分段, 分割 (4C)
 隔离; 分开 (3B)

self-contained /ˌselfkən'teɪnd/ a.	自给的, 独立自足的 (10C)
semantic /sɪ'mæntɪk/ a.	语义 (学) 的 (9B)
Semantic Grid	语义网格 (9B)
Semantic Web	语义网 (9B)
semantically /sɪ'mæntɪkəli/ ad.	在语义上 (3B)
semantics /sɪ'mæntɪks/ n.	语义 (学) (10C)
sensor /'sensə/ n.	传感器; 敏感元件 (1B)
sensor network	传感器网络, 传感网, 感知网 (6C)
sensory /'sensəri/ a.	感觉的; 感官的 (1B)
sequence number	(顺) 序号 (8B)
sequential /sɪ'kwɛnʃəl/ a.	顺序的 (4C)
serial /'siəriəl/ a.	串行的; 连续的 (2A)
server /'sɜ:və/ n.	服务器; 服务程序 (1C)
server cluster	服务器集群 (9A)
server farm	大型机服务器 (9A)
session /'seʃən/ n.	对话 (期), 会话 (4C)
severity /sɪ'verɪti/ n.	剧烈; 严重; 严厉 (10C)
sexual orientation	(对同性或异性的) 性取向, 性倾向 (11A)
shaft /ʃɑ:ft/ n.	轴 (1A)
shared lock	共享锁 (6B)
shell /ʃel/ n.	外壳; 命令解释程序; 对象处理程序 (2B)
shipping /'ʃɪpɪŋ/ n.	运送, 运输; 航运业 (8B)
shoplifting /'ʃɒplɪftɪŋ/ n.	(混在顾客群中进行的) 商店货物扒窃 (9B)
shopping cart	购物车 (12B)
shopping mall	(同一建筑物里有许多商店的) 购物中心 (9B)
Short Message Service	短信 (服务) (9C)
showbiz /'ʃəʊbiz/ n.	〈口〉娱乐性行业; 娱乐界 (= show business) (8C)
shuffle /'ʃʌfl/ v.	混洗, 洗牌 (并行处理中的一种操作) (2B)
shut-in /'ʃʌtɪn/ n.	不能外出的人; 卧病在家的人 (12C)
signature /'sɪɡnətʃə/ n.	(人或物的) 识别标志; 鲜明特征 (10B)
signature file	签名文件 (11A)
simulate /'sɪmjuleɪt/ v.	模拟, 仿真 (1C)
simulation /ˌsɪmjʊ'leɪʃən/ n.	模拟, 仿真 (1C)
simultaneous /ˌsɪmə'teɪniəs; saɪ-/ a.	同时 (发生) 的 (1C)
site map	网站地图 (11B)
skip /skip/ v.	略过, 跳过 (4B)
slack /slæk/ n.	松弛 (部分) (7B)
slash /slæʃ/ n.	斜线, 斜杠 (2B)
sloppy /'slɒpi/ a.	〈口〉马虎的; 凌乱的 (11A)
slot /slɒt/ n.	狭长孔, 狭缝 (11C)
slot machine	吃角子老虎 (一种投硬币的赌具) (11C)
small-time /'smɔ:l'taɪm/ a.	〈口〉次要的, 无关紧要的 (11C)
smart card	智能卡 (9B)
smartcard /'smɑ:tka:d/ n.	智能卡 (9C)
smartphone /'smɑ:tfeun/ n.	智能手机 (1C)
smirk /smɜ:k/ v.	假笑; 得意地笑 (11A)
snapshot /'snæpʃɒt/ n.	快照; 瞬象, 瞬态图; 抽点打印 (6B)
snoop /snu:p/ v.	窥探; 打探 (11C)
social network	社交网络 (6C)
social networking	社交网络 (9A)

social security number	社会安全(或保险、保障)号(码)(6A)
societal /sə'saɪəl/ a.	社会的(9B)
socket /'sɒkɪt/ n.	插座; 插槽; 套接字(12C)
software agent	软件代理(9B)
software overhead	软件开销(4C)
software package	软件包(3B)
software routine	软件例程(8B)
software suite	软件套件(12B)
sophistication /sə,fɪstɪ'keɪʃən/ n.	复杂性; 尖端性(3A)
sound card	声卡(1C)
source code	源(代)码(3B)
source file	源文件(4A)
source program	源程序(4A)
space /speɪs/ n.	空格, 空白(8C)
spacing /'speɪsɪŋ/ n.	间隔(1A)
spam /spæm/ n. & v.	垃圾邮件(11A) / (向...) 发送垃圾邮件(11A)
spammer /'spæmə/ n.	垃圾邮件发送者(11C)
sparing /'speəriŋ/ a.	节约的; 有节制的(11A)
spatial /'speɪʃəl/ a.	空间的(9B)
spatiotemporal /,speɪʃiəʊ'tempərəl/ a.	时空的(6C)
spec /spek/ n.	〈口〉明确说明; 规格, 规范(= specification)(5B)
-specific /spɪ'sɪfɪk/ comb. form	表示“限定的”, “特有的”(4C)
specification /,spesɪfɪ'keɪʃən/ n.	明确说明; [常作~s] 规格, 规范(1B)
spectrum /'spektrəm/ n.	频谱; 系列; 范围(9B)
speculative /'spekjuleɪtɪv, -lei/ a.	推测的; 纯理论的; 投机性的(1B)
splash page	醒目页面, 过渡页, 欢迎页面(11B)
spoof /spu:f/ v.	哄骗, 欺骗(3B)
spreadsheet /'spredʃi:t/ n.	电子表格, 电子数据表(1C)
spreadsheet program	电子数据表程序(11A)
sprout /spraut/ v.	发芽; 长出(1B)
spyware /'spaiwɛə/ n.	间谍软件(10B)
square bracket	方括号(3C)
star topology	星型拓扑结构(7A)
state of the art	(学科、技术等当前的或某一时期的) 发展水平, 最新水平(10A)
stateless system	无状态系统(12B)
statement /'steɪtmənt/ n.	语句(3A)
statistician /,stætɪ'stɪʃən/ n.	统计学家; 统计员(1A)
status word	状态字(2B)
stealthy /'stelθi/ a.	偷偷摸摸的, 暗中进行的, 秘密的(10C)
stock quote	股票行情(4C)
stock-keeping unit	库存单位(12B)
storage register	存储寄存器(3A)
storage server	存储服务器(1C)
store /sto:/ n.	存储(器)(2C)
storefront /'sto:frʌnt/ n.	店面, 铺面(12C)
straightforward /,streɪt'fɔ:wəd/ a.	径直的; 正直的; 简单的(2C)
strategically /stra'teɪdʒɪkəli/ ad.	在战略上; 策略地(11B)
stream data	流数据(6C)
strive /straɪv/ v.	努力; 力争(1B)

structure chart	结构图 (4B)
stylus /'stajləs/ n.	([复] -luses 或 -li / -lai /) 输入笔, 光笔 (2A)
subconsciously /,sʌb'kɒnʃəsli/ ad.	下意识地, 潜意识地 (11B)
subdirectory /,sʌbdɪ'rektəri/ n.	子目录 (2B)
submarine /,sʌbmə'ri:n/ n.	潜艇 (1A)
subnetting /'sʌb.netɪŋ/ n.	子网划分, 子网组建 (7B)
subroutine /'sʌbru:ti:n/ n.	子例程, 子(例行)程序 (4A)
subscriber /səb'skraɪbə/ n.	用户, 订户 (9C)
subscript /'sʌbskript/ n.	下标, 角注 (3C)
subscription /səb'skriptsən/ n.	订购; 订阅 (9A)
substrate /'sʌbstreɪt/ n.	衬底, 基底 (1A)
subvert /səb'vɜ:t, sʌb-/ v.	颠覆; (暗中)破坏; 搅乱 (2C)
sue /sju:; su:/ v.	控告, 起诉 (11C)
suffix /'sʌfiks/ n.	后缀 (8A)
suite /swi:t/ n.	(同类物的)系列, 组, 套; (软件的)套件 (7A)
supercomputer /'sju:pəkəmputə/ n.	超级计算机, 巨型计算机 (1A)
superconductivity /'sju:pəkɒndʌk'tiviti/ n.	超导(电)性 (1A)
superfluous /sju:pə'fluəs/ a.	多余的; 过剩的 (3B)
supervisor /'sju:pəvaɪzə/ n.	监督(人); 管理人; 指导者 (11A)
supply chain	供应链 (9B)
suppress /sə'pres/ v.	压制; 抑制; 阻止 (10A)
suppression /sə'presʃən/ n.	压制; 抑制 (9B)
surf /sɜ:f/ v.	(在...)冲浪 (9C)
surveillance /sə'veɪləns, sə-/ n.	监视 (6C)
susceptible /sə'septəbl/ a.	易受影响的 (to) (10A)
suspicious /sə'spiʃəs/ a.	可疑的 (10B)
swap /swɒp/ v.	交换 (10C)
switch /'swɪtʃ/ n.	交换(设备); 开关; 转换 (1A)
symptomatic /,sɪmptə'mætɪk/ a.	作为征候(或征兆)的; 表明的 (of) (5C)
synchronization /,sɪŋkrənaɪ'zeɪʃən/ n.	同步(化) (3B)
synchronize /'sɪŋkrənaɪz/ v.	(使)同步; (使)协调 (1C)
synchronous /'sɪŋkrənəs/ a.	同时的; 同步的 (4C)
synchrony /'sɪŋkrəni/ n.	同时(性); 同步(性) (2C)
synergistic /,sɪnə'dʒɪstɪk/ a.	增效的; 协同作用的; 协作的 (9B)
synonym /'sɪnənim/ n.	同义词 (6C)
syntactic /sɪn'tæktɪk/ a.	(按照)句法的 (3B)
syntax /'sɪntæks/ n.	句法 (3A)
system call	系统调用 (4A)
system crash	系统崩溃 (10A)
system integration	系统集成 (4C)
system routine	系统例(行)程(序) (4A)

T

tablet computer	平板电脑 (9B)
tack /tæk/ v.	附加, 增补, 追加 (on) (9C)
tangible /'tæŋdʒəbl/ a.	可触摸的; 有形的; 明确的 (9B)
tapestry /'tæpɪstri/ n.	花毯; 挂毯 (4A)
targeted /'tɑ:ɡɪtɪd/ a.	有目标的; 指向目标的; 被定为目标的 (8C)

technicality /ˈteknɪˈkælɪti/ n.	技术性(细节)(1C)
technologist /ˈteknələdʒɪst/ n.	技术专家;〈美〉技术员, 技师(6C)
telecom /ˈtelɪkɒm/ n.	电信(= telecommunication)(9B)
telemedicine /ˈteliˌmedɪsɪn/ n.	远程医学(9A)
teleoperation /ˌtelɪɒpəˈreɪʃən/ n.	远程操作, 遥操作(9B)
telepresence /ˈteliˌprezəns/ n.	远程呈现(9B)
telnet /ˈtelnet/ n.	[亦作 Telnets] 远程登录协议(或程序、服务)(8B)
template /ˈtemplɪt/ n.	模板(5C)
tenant /ˈtenənt/ n.	租户, 承租人(9A)
terabyte /ˈterəbaɪt/ n.	太字节, 万亿字节(6C)
terminate /ˈtɜːmɪneɪt/ v.	(使)终止(2B)
termination /ˌtɜːmɪˈneɪʃən/ n.	结束, 终止(4C)
terminator /ˈtɜːmɪneɪtə/ n.	终结器; 端子; 终结符(7B)
terminology /ˌtɜːmɪˈnɒlədʒi/ v.	[总称] 术语(7A)
test harness	测试框架(5B)
test script	测试脚本(5B)
testbed /ˈtestbed/ n.	试验台, 测试台(5B)
textbox /ˈtekstbɒks/ n.	文本(或正文、文字)框(11A)
text box	文本框(12B)
text editor	文本编辑程序, 文本编辑器(4A)
texting /ˈtekstɪŋ/ n.	发短信(9C)
textual /ˈtekstʃuəl, -tʃuəl/ a.	文本的, 正文的(2B)
theorem /ˈθiərəm/ n.	定理(3B)
theorem prover	定理证明程序, 定理证明器(3B)
thermostat /ˈθɜːməʊstæt/ n.	温度自动调节器, 恒温器(9B)
thread /θred/ n.	线程, 线索(3B)
three-dimensional /ˈθriːdiˈmenʃənəl/ a.	三维的, 立体的(1B)
throughput /ˈθruːput/ n.	吞吐量(9A)
throwaway /ˈθrəʊəˌwei/ a.	使用后抛弃的, 一次性使用的(5A)
thrust /θrʌst/ n.	要点, 要旨; 目标(10A)
tier /tiə/ n.	(一)层(2C)
tightly coupled system	紧密耦合系统(4C)
time slice	时间片(2B)
timer /ˈtaɪmə/ n.	计时器, 定时器; 定时程序(2B)
time-sharing /ˈtaɪmˌʃeərɪŋ/ n. & a.	分时(技术或方法)(的)(2B)
token /ˈtəʊkən/ n.	权标, 令牌(7A)
token ring	权标环, 令牌环(7A)
toner /ˈtəʊnə/ n.	色粉, 墨粉(2A)
toolset /ˈtuːlset/ n.	成套工具, 工具箱(2C)
top-level domain	顶级域名(8A)
topology /ˈtəʊpələdʒi, tɒ-/ n.	拓扑(结构), 布局(7A)
tornado /ˈtɔːneɪdə/ n.	龙卷风(1C)
torpedo /ˈtɔːpiːdə/ n.	鱼雷(1A)
touch screen	触摸屏, 触屏(1C)
touch-sensitive /ˈtʌtʃˌsensɪtɪv/ a.	触敏的(1C)
touch-sensitive screen	触摸屏(1C)
trackball /ˈtrækbɔːl/ n.	跟踪球, 轨迹球(2A)
tradeoff /ˈtreɪdɒf/ n.	平衡, 权衡; (公平)交易(5C)
trafficwise /ˈtræfɪkwaɪz/ ad.	在交通方面(7C)
tragic(al) /ˈtrædʒɪk(əl)/ a.	悲剧(性)的; 悲惨的(5C)

transaction /træn'zækʃən, -'sæk-/ n.	交易; 事务(处理), 事项(处理) (1A)
transient /'trænzient/ a.	短暂的; 临时的 (4C)
transistor /træn'zistə, -'sis-/ n.	晶体管 (1A)
transit /'trænsit, -zit/ n.	运输, 运送 (12C)
translator /træns'leite/ n.	翻译程序, 翻译器 (2C)
transmitter /trænz'mitə; træns-/ n.	发射机; 发送器 (8A)
trapdoor /'træpdɔ:/ n.	陷阱门, 天窗 (10C)
triad /'traid, -æd/ n.	三件一套(或组); 三位一体 (10A)
trigger /'trige/ v. & n.	触发 (2C) / 引爆器; 触发器 (10C)
Trojan /'trəudʒən/ (horse) n.	特洛伊木马程序(或病毒) (3B)
trust delegation	信任委托 (9A)
trustmark /'trastmɑ:k/ n.	信任标记 (12C)
trustworthiness /'trast,wə:ðinis/ n.	值得信任; 可信; 可靠 (4C)
trustworthy /'trast,wə:ði/ a.	值得信任的; 可信的; 可靠的 (3B)
tuple /'tʌpl, 'tju:pl/ n.	元组, 字节组 (6A)
tussle /'tʌsəl/ n.	争执, 争辩 (11C)
tweet /twi:t/ v.	上推特, 发微博 (9C)
two-dimensional array	二维数组 (3C)
two-dimensional table	二维表 (6A)
typically /'tipikəli/ ad.	一般地, 通常 (1C)
typo /'taipəu/ n.	打字(或排印)错误 (11A)

U

ubiquitous /ju:'bikwites/ a.	普遍存在的, 无所不在的 (6C)
unattended /ʌnə'tendid/ a.	没人看管的; 没人照料的 (9B)
unauthorized /ʌn'ɔ:θəraizd/ a.	未经授权的 (3B)
underlie /ʌndə'lai/ v.	构成...的基础(或起因); 支持 (1B)
underline /'ʌndə,lain/ n.	下划线 (8C)
underlying /ʌndə'laɪɪŋ/ a.	基本的, 根本的 (2C)
undertaking /ʌndə'teɪkɪŋ, 'ʌndet-/ n.	任务; 事业 (8A)
underway /'ʌndə'wei/ a.	在进行中的 (3B)
unforeseen /ʌnfo:'si:n/ a.	未预见到的; 意料之外的 (1B)
unified /'ju:nifaɪd/ a.	统一的 (6C)
Uniform Resource Locator	统一资源定位符, 统一资源定位器 (8C)
unify /'ju:nifai/ v.	使联合, 统一; 使一致 (1B)
unimpaired /ʌnim'peəd/ a.	未受损的; 未削弱的 (10A)
union /'ju:njən/ n.	共用体(定义关键字), 共用(数据类型) (3B)
unit testing	单元测试 (5A)
unscrupulous /ʌn'skru:pjʊləs/ a.	肆无忌惮的; 无耻的 (10C)
unstructured /ʌn'strʌktʃəd/ a.	非结构化的, 没有特定结构(或系统)的, 松散的 (12A)
unsubscribe /ʌnsəb'skraib/ v.	退订, 取消订购(或订阅) (10B)
update /'ʌpdeɪt/ n.	更新; 修改 (6B)
update /ʌp'deɪt/ v.	更新; 修改 (2B)
upgrade /ʌp'greɪd, 'ʌpg-/ v.	使升级; 改善 (2C)
upload /ʌp'ləʊd/ v.	上传, 上载 (9C)
uppercase /'ʌpə'keɪs/ a.	(字母)大写的 (8A)

use case
 user ID
 utility /ju:'tiliti/ n.
 utility package
 utility program
 utility software
 utilization /ju:'tilaɪ'zeɪʃən; -li'z-/ n.

用例 (5B)
 用户标识 (符) (10C)
 实用程序, 公用程序 (4A)
 实用软件包, 公用程序包 (7A)
 实用程序, 公用程序 (4A)
 实用软件 (8B)
 利用 (9A)

V

vacuum tube
 validate /'vælɪdeɪt/ v.
 validation /ˌvæli'deɪʃən/ n.
 validity /və'lɪdɪti/ n.
 vector /'vektə/ n.
 vend /vend/ v.
 vender /'vendə/ n.
 vending machine
 vendor /'vendə/ n.
 verbal /'vɜ:bəl/ a.
 versatile /'vɜ:sə'taɪl/ a.
 versatility /ˌvɜ:sə'tɪlɪti/ n.
 versus /'vɜ:səs/ prep.
 viable /'vaɪəbl/ a.
 video display
 video game
 videogame /'vɪdiəʊgeɪm/ n.
 videogame console
 view /vju:/ n.
 vigor /'vɪɡə/ n.
 violate /'vaɪəleɪt/ v.
 viral /'vaɪrəl/ a.
 virtual /'vɜ:tʃuəl/ a.
 virtual machine
 virtual memory
 virtual reality
 virtualization /ˌvɜ:tʃuəlaɪ'zeɪʃən/ n.
 virtualize /'vɜ:tʃuəlaɪz/ v.
 virus signature
 visualization /ˌvɪzjuəlaɪ'zeɪʃən/ n.
 visualize /'vɪzjuəlaɪz/ v.
 volatile /'vɒlətaɪl/ a.
 volatile memory
 vulnerability /ˌvʌlnərə'bɪlɪti/ n.
 vulnerable /'vʌlnərəbl/ a.

真空管 (1A)
 确认 (…有效), 证实, 验证 (4B)
 确认, 验证 (5A)
 有效 (性) (10A)
 传病媒介; 矢量 (10C)
 出售 (尤指沿街叫卖零售); 用售货机出售 (9C)
 卖主; 厂家, 厂商 (7A)
 投币式自动售货机 (9C)
 卖主; 厂家, 厂商 (3B)
 用言辞的; 文字上的; 口头的 (11C)
 多用途的, 通用的 (1A)
 多用途; 通用性 (1C)
 对; 与…相对 (或相比) (10A)
 切实可行的; 可实施的 (1B)
 视频显示 (器) (2A)
 电子游戏 (2A)
 电子游戏 (1C)
 电子游戏机 (1C)
 视图 (3C)
 体力; 精力; 活力 (6C)
 违反, 违犯 (11A)
 病毒 (性) 的; 病毒引起的 (10C)
 虚拟的 (1B)
 虚拟机, 虚 (拟计算) 机 (4C)
 虚 (拟) 存 (储器), 虚拟存储 (器), 虚拟内存 (2B)
 虚拟现实 (1B)
 虚拟化 (9A)
 虚拟化 (9A)
 病毒特征代码 (10B)
 可视化, 直观化 (1B)
 使可视化, 使可见; 使直观化 (4B)
 易失的; 易变的 (2A)
 易失性存储器 (2A)
 易遭攻击的地方; 脆弱 (性) (10A)
 脆弱的; 易受攻击的 (3B)

W

wade /weɪd/ v.
walkthrough /'wɔ:kθru:/ n.
warehouse /'weəhaus/ n.
warehousing /'weə,haʊzɪŋ/ n.
warrant /'wɒrənt; 'wɔ:-/ v. & n.

waterfall /'wɔ:təfɔ:l/ n.
waterfall model
wearable computer
Web browser
Web community
Web page
Web server
Web site
web-authoring software
webmaster /'web,mɑ:stə/ n.
webpage /'webpeɪdʒ/ n.
website /'websaɪt/ n.
whereby /'hweə'baɪ/ ad.
white box testing
whiteboard /'hwaɪtbɔ:d/ n.
whitelist /'hwaɪtlɪst/ n.
whitelisting /'hwaɪt'lɪstɪŋ/ n.
wide area network
wildcard /'waɪldkɑ:d/ n.
wildcard character
wink /wɪŋk/ v.
wire transfer
witty /'wɪti/ a.
word processor
workflow /'we:kfləʊ/ n.
workload /'we:kleʊd/ n.
workstation /'we:k,steɪʃən/ n.
worm /wɜ:m/ n.
wristwatch /'rɪstwɒtʃ/ n.

跋涉, 艰难地行走 (11B)
走查 (5B)
仓库, 货栈 (6C)
存库, 入栈 (6C)
证明...是正当 (或有理);
应受 (6C) / 授权 (令) (11C)
瀑布 (4B)
瀑布模型 (5A)
可穿戴计算机, 穿戴式计算机 (9C)
网络浏览器 (2C)
Web 社区, 网络社区 (6C)
(万维) 网页 (1C)
万维网服务器 (8C)
网站, 站点 (1C)
网络写作软件 (11B)
网站管理员 (11B)
网页 (11A)
网站, 站点 (11A)
靠那个; 借以 (9A)
白盒测试 (法) (5B)
白色书写板 (11B)
白名单 (10B)
白名单编制 (10B)
广域网 (6A)
通配符, 万能符 (8C)
通配符 (8C)
眨眼; 眨眼示意, 使眼色 (11A)
电汇 (12C)
诙谐的; 说话风趣的 (11A)
文字处理软件 (11A)
工作流 (9B)
工作量; 工作负荷 (9A)
工作站 (1A)
蠕虫 (病毒) (10B)
手表 (1A)

Z

zebra /'zi:brə, 'ze-/ n.
zombie /'zɒmbi/ n.

斑马 (9C)
还魂尸, 僵尸; 僵 (进程) (10C)

Abbreviations

(缩略语表)

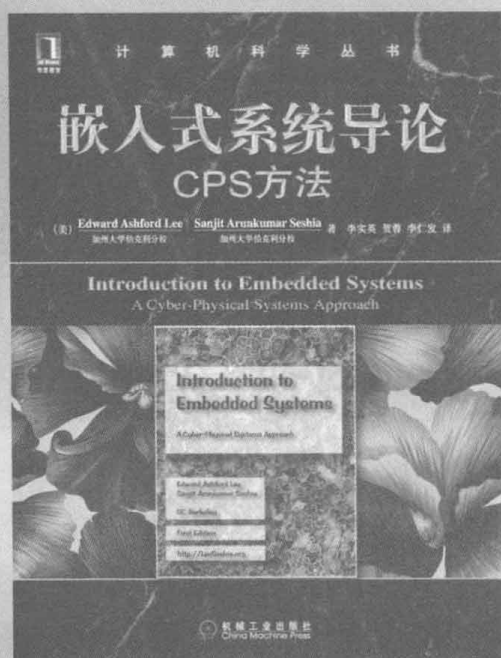
3G	<i>3rd Generation</i>	第三代移动通信技术 (9C)
4G	<i>4th Generation</i>	第四代移动通信技术 (9C)
ABC	<i>American Broadcasting Company</i>	美国广播公司 (8C)
AI	<i>artificial intelligence</i>	人工智能 (1B)
ALGOL	<i>Algorithmic Language</i>	ALGOL 语言, 算法语言 (3A)
ANSI	<i>American National Standards Institute</i>	美国国家标准协会 (4C)
ANSI/SPARC	<i>American National Standards Institute Standards Planning and Requirements Committee</i>	美国国家标准协会标准计划与需求委员会 (6A)
API	<i>Application Program Interface</i>	应用程序接口 (3B)
APSE	<i>Ada Programming Support Environment</i>	Ada 程序设计支持环境 (2C)
AWS	<i>Amazon Web Services</i>	亚马逊网络服务 (9A)
B2B	<i>business-to-business</i>	企业对企业 (12A)
B2C	<i>business-to-consumer</i>	企业对消费者 (12A)
B2G	<i>business-to-government</i>	企业对政府 (12A)
BASIC	<i>Beginner's All-purpose Symbolic Instruction Code</i>	BASIC 语言, 初学者通用符号指令码 (3A)
bcc	<i>blind carbon copy</i>	密送 (11A)
BIOS	<i>Basic Input/Output System</i>	基本输入 / 输出系统 (2A)
C2C	<i>consumer-to-consumer</i>	消费者对消费者 (12A)
CAD	<i>computer-aided design</i>	计算机辅助设计 (2C)
CAPTCHA	<i>Completely Automated Public Turing Test to Tell Computers and Humans Apart</i>	全自动区分计算机和人类的图灵测试, 验证码 (11C)
CASE	<i>computer-aided software engineering</i>	计算机辅助软件工程 (2C)
CBSE	<i>component-based software engineering</i>	基于组件的软件工程 (5A)
cc	<i>carbon copy</i>	抄送 (11A)
CD	<i>compact disc</i>	光盘 (12C)
CD-ROM	<i>compact disc read-only memory</i>	只读光盘 (存储器) (2A)
CIA	<i>confidentiality, integrity, and availability</i>	机密性、完整性与可用性 (10A)
CLOS	<i>Common Lisp Object System</i>	公共 Lisp 对象系统 (5C)
CNN	<i>Cable News Network</i>	(美国) 有线电视新闻网 (8C)
COBOL	<i>Common Business-Oriented Language</i>	COBOL 语言, 面向商业的通用语言 (3A)
COD	<i>cash on delivery</i>	货到付款, 交货付款 (12C)

CORBA	Common Object Request Broker Architecture	公用对象请求代理(程序)体系结构(3B)
COTS	commercial off-the-shelf	现成商用的, 现成民用的(5A)
CPU	central processing unit	中央处理器(1C)
CRM	customer relationship management	客户关系管理(12B)
CRT	cathode ray tube	阴极射线管(2A)
CSMA-CD	Carrier Sense Multiple Access with Collision Detection	带有冲突检测的载波侦听多路访问(协议)(7A)
DARPA	Defense Advanced Research Projects Agency	(美国)国防部高级研究计划局(8A)
DBA	database administrator	数据库管理员(6A)
DBMS	database management system	数据库管理系统(6A)
DCS	Digital Collection System	数字收集系统(11C)
DMCA	Digital Millennium Copyright Act	《数字千年版权法》,《千禧年数字版权法》(11C)
DNA	deoxyribonucleic acid	脱氧核糖核酸(6C)
DNS	domain name system	域名系统(8A)
DSL	Digital Subscriber Line	数字用户线路(8A)
DVD	digital video disc	数字(激光)视盘(1C)
EDI	electronic data interchange	电子数据交换(12A)
ENIAC	Electronic Numerical Integrator And Computer	电子数字积分计算机, ENIAC 计算机(1A)
ERM	entity-relationship model	实体关系模型(6A)
ERP	enterprise resource planning	企业资源计划(12B)
FAST	Framework for the Application of Systems Techniques	系统技术应用框架(4B)
FBI	Federal Bureau of Investigation	(美国)联邦调查局(11C)
FM	frequency modulation	调频(广播系统), 频率调制(1C)
FORTRAN	Formula Translation	FORTRAN 语言, 公式翻译程序语言(3A)
FTP	File Transfer Protocol	文件传送协议(8B)
GAE	Google App Engine	谷歌应用引擎(9A)
gif	Graphics Interchange Format	图形交换格式(11B)
GPS	Global Positioning System	全球(卫星)定位系统(1C)
GUI	graphical user interface	图形用户界面(2B)
HTC	high-throughput computing	高吞吐(量)计算(9A)
HTML	Hypertext Markup Language	超(级)文本标记语言(8C)
HTTP, http	Hypertext Transfer Protocol	超(级)文本传送(或传输)协议(2C)
HTTPS	Hypertext Transfer Protocol over Secure Sockets Layer	(使用)安全(套接层的)超(级)文本传送(或传输)协议(12C)
IBM	International Business Machines	(美国)国际商用机器公司(1C)
IC	integrated circuit	集成电路(1A)

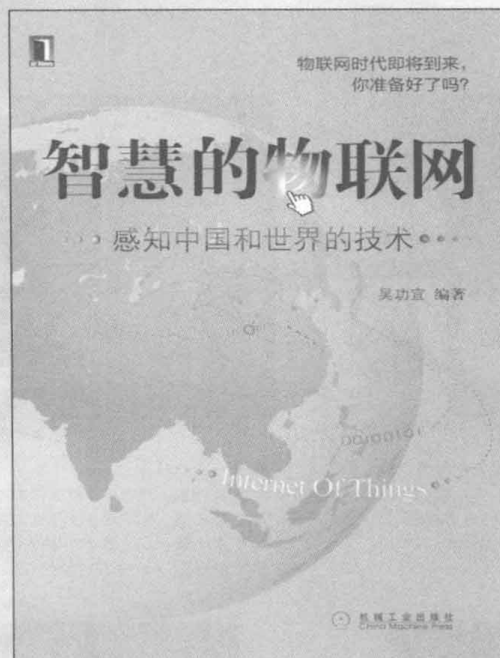
ICANN	<i>Internet Corporation for Assigned Names and Numbers</i>	互联网名称与数字地址分配机构 (8A)
ICL	<i>International Computers Limited</i>	国际计算机有限公司 (2C)
ID	<i>identification</i>	身份证明, 身份识别, 标识 (10C)
IE	<i>information engineering</i>	信息工程 (4B)
IE	<i>Internet Explorer</i>	(微软) 因特网浏览器 (8C)
IoT	<i>Internet of Things</i>	物联网 (9B)
IP	<i>Internet Protocol</i>	IP 协议, 网际协议, 网间协议 (8A)
IPv6	<i>Internet Protocol version 6</i>	IP 协议第 6 版 (9B)
ISO	<i>International Organization for Standardization</i>	国际标准化组织 (4C)
ISP	<i>Internet service provider</i>	因特网服务提供商 (8A)
IT	<i>information technology</i>	信息技术 (9A)
jpeg	<i>Joint Photographic Experts Group</i>	联合图像专家小组 (11B)
KB	<i>kilobyte</i>	千字节 (2B)
KDD	<i>knowledge discovery from data</i>	从数据中发现知识 (6C)
LAN	<i>local area network</i>	局域网 (7A)
LCD	<i>liquid crystal display</i>	液晶显示 (器) (2A)
LISP	<i>List Processing</i>	LISP 语言, 表处理语言 (3A)
LSI	<i>large-scale integrated; large-scale integration</i>	大规模集成的; 大规模集成 (电路) (1A)
MAN	<i>metropolitan area network</i>	城域网 (7A)
MB	<i>megabyte</i>	兆字节 (2B)
MDD	<i>model-driven development</i>	模型驱动开发 (4B)
NFC	<i>Near Field Communication</i>	近场通信, 近距离无线通信 (9C)
NIST	<i>National Institute of Standards and Technology</i>	(美国) 国家标准与技术研究院 (10A)
OMG	<i>Object Management Group</i>	对象管理组 (4C)
OOAD	<i>object-oriented analysis and design</i>	面向对象分析与设计 (4B)
OSI	<i>Open Systems Interconnection</i>	开放式系统互连 (2C)
P2P	<i>peer-to-peer</i>	对等的 (7A)
PC	<i>personal computer</i>	个人计算机, 个人电脑 (1C)
PDA	<i>personal digital assistant</i>	个人数字助理 (1C)
PROLOG	<i>Programming in Logic</i>	PROLOG 语言, 逻辑程序设计语言 (3A)
QoS	<i>Quality of Service</i>	服务质量 (9A)
RAM	<i>random-access memory</i>	随机 (存取) 存储器 (2A)
RC2	<i>Research Compute Cloud</i>	研究计算云 (9A)
RDBMS	<i>relational database management system</i>	关系数据库管理系统 (6A)
RF	<i>radio frequency</i>	射频; 无线电频率 (9B)
RFID	<i>radio-frequency identification</i>	射频识别 (9B)

ROM	<i>read-only memory</i>	只读存储器 (1A)
SaaS	<i>Software as a Service</i>	软件即服务 (9A)
SAN	<i>storage area network</i>	存储区域网 (络) (9A)
SCM	<i>supply chain management</i>	供应链管理 (12B)
SF	<i>science fiction</i>	科学幻想小说, 科幻小说 (10C)
SIMULA	<i>Simulation Language</i>	SIMULA 语言, 仿真语言, 模拟语言 (3A)
SKU	<i>stock-keeping unit</i>	库存单位 (12B)
SLA	<i>service-level agreement</i>	服务水平 (或级别、等级) 协议 (9A)
SMS	<i>Short Message Service</i>	短信服务 (12C)
SOA	<i>service-oriented architecture</i>	面向服务的体系结构, 面向服务架构 (9A)
SQL	<i>Structured Query Language</i>	结构化查询语言 (3A)
SSL	<i>Secure Sockets Layer</i>	安全套接层 (协议) (12C)
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>	TCP/IP 协议, 传输控制 (协议) / 网际 (或网间) 协议 (4C)
TLD	<i>top-level domain</i>	顶级域名 (8A)
UK	<i>United Kingdom</i>	联合王国, 英国 (2C)
URL	<i>Uniform Resource Locator</i>	统一资源定位符, 统一资源定位器 (8B)
VLSI	<i>very large-scale integrated; very large-scale integration</i>	超大规模集成的; 超大规模集成 (电路) (1A)
VM	<i>virtual machine</i>	虚拟机 (9A)
WAN	<i>wide area network</i>	广域网 (7A)
Wi-Fi	<i>Wireless Fidelity</i>	无线保真 (9B)
WSN	<i>wireless sensor network</i>	无线传感网 (9B)
WWW	<i>World Wide Web</i>	万维网 (2C)
XML	<i>eXtensible Markup Language</i>	可扩展标记语言 (2C)

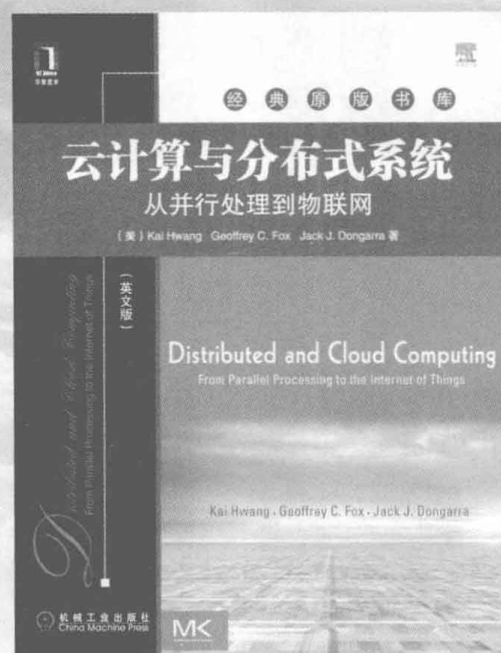
相关图书



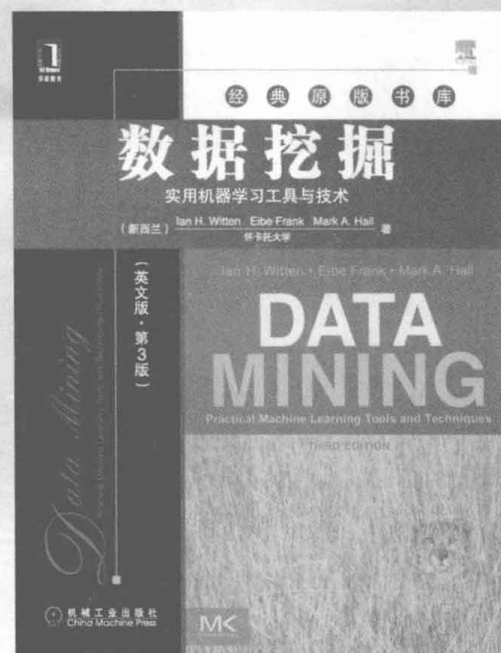
书号：978-7-111-36021-6
定价：55.00元



书号：978-7-111-30710-5
定价：36.00元



书号：978-7-111-38227-0
定价：99.00元



书号：978-7-111-37417-6
定价：108.00元

机械工业出版社—华章计算机经典外版丛书新品推荐



作者: Randal E. Bryant 等著
中文翻译版: 978-7-111-32133-0, 99.00元
英文影印版: 978-7-111-32631-1, 128.00元
本科教学版: 2013年出版



作者: Y. Daniel Liang
中文基础篇: 978-7-111-34081-2, 75.00元
英文影印版: 978-7-111-36122-0, 89.00元
中文进阶篇: 978-7-111-34236-6, 79.00元
英文影印版: 978-7-111-36125-1, 89.00元



英文影印版: 978-7-111-32674-8, 78.00元
中文翻译版: 978-7-111-25121-7, 89.00元
本科教学版: 978-7-111-26929-8, 55.00元



作者: Kenneth H. Rosen
中文翻译版: 978-7-111-35039-2, 99.00元
英文影印版: 978-7-111-23935-2, 89.00元
英文精编版: 978-7-111-31329-8, 55.00元
本科教学版: 978-7-111-35950-0, 49.00元



作者: Roger S. Pressman
中文翻译版: 978-7-111-33581-8, 79.00元
英文影印版: 978-7-111-31871-2, 75.00元
本科教学版: 978-7-111-35350-8, 55.00元
英文精编版: 978-7-111-35965-4, 49.00元



作者: Ian Sommerville
中文翻译版: 978-7-111-33498-9, 75.00元
英文影印版: 978-7-111-34825-2, 99.00元



作者: Stephen R. Schach
中文翻译版: 978-7-111-36273-9, 65.00元
英文影印版: 978-7-111-34196-3, 79.00元



作者: Bob Hughes 等著
中文翻译版: 978-7-111-30964-2, 39.00元
英文影印版: 978-7-111-30537-8, 39.00元



作者: June Jamrich Parsons 等
中文翻译版: 978-7-111-32655-7, 75.00元
英文影印版: 978-7-111-31799-9, 75.00元



作者: David A. Patterson 等
中文翻译版: 978-7-111-35305-8, 99.00元
英文影印版 (ARM版): 978-7-111-30288-9, 95.00元



作者: Brian W. Kernighan 等著
中文翻译版: 978-7-111-12806-0, 30.00元
英文影印版: 978-7-111-19626-0, 35.00元



作者: Jeffrey L. Whitten
中文翻译版: 978-7-111-36386-6, 49.00元
英文影印版: 978-7-111-35278-5, 79.00元

Images have been losslessly embedded. Information about the original file can be found in PDF attachments. Some stats (more in the PDF attachments):

```
{
  "filename": "XzEzMjQ3Njg1LnppcA==",
  "filename_decoded": "_13247685.zip",
  "filesize": 59301332,
  "md5": "7eb099e96d3f8862f3fde6fa0001b4f0",
  "header_md5": "6c9a12f501829763129c256945dd326c",
  "sha1": "1f5076e477868ae5311872586632d3e6d17953fb",
  "sha256": "c9fb6bc4bd5fcb2c34ed74d594568b1b56434b402404b432d882d8fc4e802ef2",
  "crc32": 2322212545,
  "zip_password": "",
  "uncompressed_size": 64042894,
  "pdg_dir_name": "_13247685",
  "pdg_main_pages_found": 316,
  "pdg_main_pages_max": 316,
  "total_pages": 324,
  "total_pixels": 2068268108,
  "pdf_generation_missing_pages": false
}
```