

COMPUTER ENGLISH (SE)

计算机英语 (第2版)

刘艺 王春生 编



计算机英语

Computer (第2版)

(Second Edition)

English

刘艺 王春生 编



机械工业出版社
China Machine Press

内容提要

本书是一本面向 21 世纪的计算机专业英语教材，它涉及了计算机基础知识、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的生活信息技术。全书以计算机和 IT 领域的最新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英语的一般特点和大量专业词汇，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，特别是版式设计创意新颖，极大方便了读者学习和查阅。书后还附有词汇表和缩略语表。

本书可供大专院校计算机及 IT 相关专业的学生使用，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谋求出国发展的计算机人才学习参考。

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。有些院校甚至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。全书选材广泛，内容丰富，涉及了计算机基础知识、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。

本书第一版作为 IT 专业英语教材出版以来深受读者喜爱，被全国众多高校采用。短短三年时间里，印数已超过 3 万册。

鉴于全球 IT 业发展迅速、变化频繁，新技术和新术语大量涌现。为适应这种情况，深化教学改革，我们对第一版教材进行了全面修订，推出《计算机英语 第 2 版》。这次修订主要做了以下工作：

一、调整了部分课文 第 2 版替换了 9 篇内容过时或不太合适的课文。新增的课文更加紧扣技术潮流，把握主流经典。既有涉及微软的 .NET 和 SUN 的 J2EE 两大技术阵营的最新技术时文，也有《设计模式》这样经典原著，还有对下一代操作系统 Longhorn 的展望。并由此给读者带来了大量新鲜的技术术语和缩略语，为读者尽快融入英文技术社区，阅读原文资料提供帮助。

二、对保留的课文进行了修改 借这次修订，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。

三、优化了全书的体系结构 原书第 1 版共有 15 个单元，现调整为 14 个单元，避免内容过多，在有限的教学时数中难以完成。我们删除了原书第 1 版的 Distributed Systems（分布式系统）单元、新增了 Software Engineering（软件工程）单元，并将原来的 Network（网络）和 Network Infrastructure（网络基础结构）两个单元合并成一个 Computer Networks（计算机网络）单元。同时在全书的体系结构编排上使之更符合计算机科学的体系结构，基本上遵循了计算机基础、硬件、软件、网络与通信、应用等五个层次。

四、调整了版式 我们在保留原书最具特色的旁注生词、脚注难点的同步对照注释版式外，还通过套色印刷，使生词更加醒目，加强读者对计算机词汇的理解和记忆。另外，书后还附有生词表和缩略语表，而且所有生词与缩略语均注明在书中首次出现的课次与具体文章。它们既可作为全书的索引，方便读者进行相关内容的查阅复习，也可作为一本实用的英汉计算机小词典，在日常学习和生活中备用。同时，我们还为大部分文章精心选配了插图和示意图，以增加文章内容的直观性与趣味性，使读者能更好地理解原文和提高学习兴趣。

五、提供完善的在线服务 为了加大对授课老师的支持力度，我们为老师准备了参考译文、练习答案、考试试卷等教辅材料，并筹备制作多媒体教学软件。授课老师可以登陆：<http://www.hzbook.com> 注册并下载配套的教辅材料。也可以直接与机械工业出版社华章分社联系：

电话：010-68995264 010-68995260

Email: chiefeditor@hzbook.com marketing@hzbook.com

对于本书的读者，还可以在线访问作者网站 <http://www.liu-yi.net> 或本书的论坛：<http://www.liu-yi.net/BBS/list.asp?boardid=3>

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，同时洪蕾、王珊珊、蒋丹丹等也为本书的编撰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，还恳请广大读者不吝赐教。

编 者

2001年5月于南京

E-mail: book4u@sina.com

使用说明

1、本教材根据其内容分为四大部分，共 15 课。每一课包括三篇文章。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。

2、课文 A 长度一般为 1400~1800 词，要求能正确理解和熟练掌握其内容。课文 B 和课文 C 长度一般为 1200~1600 词，要求能掌握中心大意，抓住主要事实。

3、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。

4、国家教育部于 1999 年新颁布的《大学英语教学大纲》（修订本）规定：“学生在完成基础阶段的学习任务，达到四级或六级后，都必须修读专业英语。”这就意味着学生在学习专业英语之前必须达到大学英语四级水平，但不一定达到六级。因此，本教材的生词选注以四级为起点，参照标准为 2000 年 7 月出版的《大学英语教学大纲》（修订本）配套词汇表。所注生词计有两类：一类是通用词，即大纲词汇表未列入 1~4 级的词汇；另一类为计算机及相关专业词汇，是大纲词汇表未列入 1~4 级，或未注明有关词义的词汇。生词均用黑体在课文中标出，并在课文对应段落旁边进行注释，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，但书后附有生词表以备查检。

5、计算机英语的特点之一是大量缩略语的使用。本教材对其采取两种处理方式：一是在首次出现处进行脚注；二是对大家十分熟知或其意思在文中不言自明的缩略语未加脚注。但是，无论那种情况，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、完成本教材的学习后应达到的要求，可参照《大学英语教学大纲》（修订本）有关专业英语的规定。

7、计算机专业英语课及其本教材的使用，本科生可安排在第五至第七学期，研究生可安排在第一学年。对于教学时数比较充裕的院校，可全书使用；对教学时数比较少的院校，重点可放在前三部分，而将第四部分作为课外阅读，也可按照自身的需求进行取舍。

编者

目 录

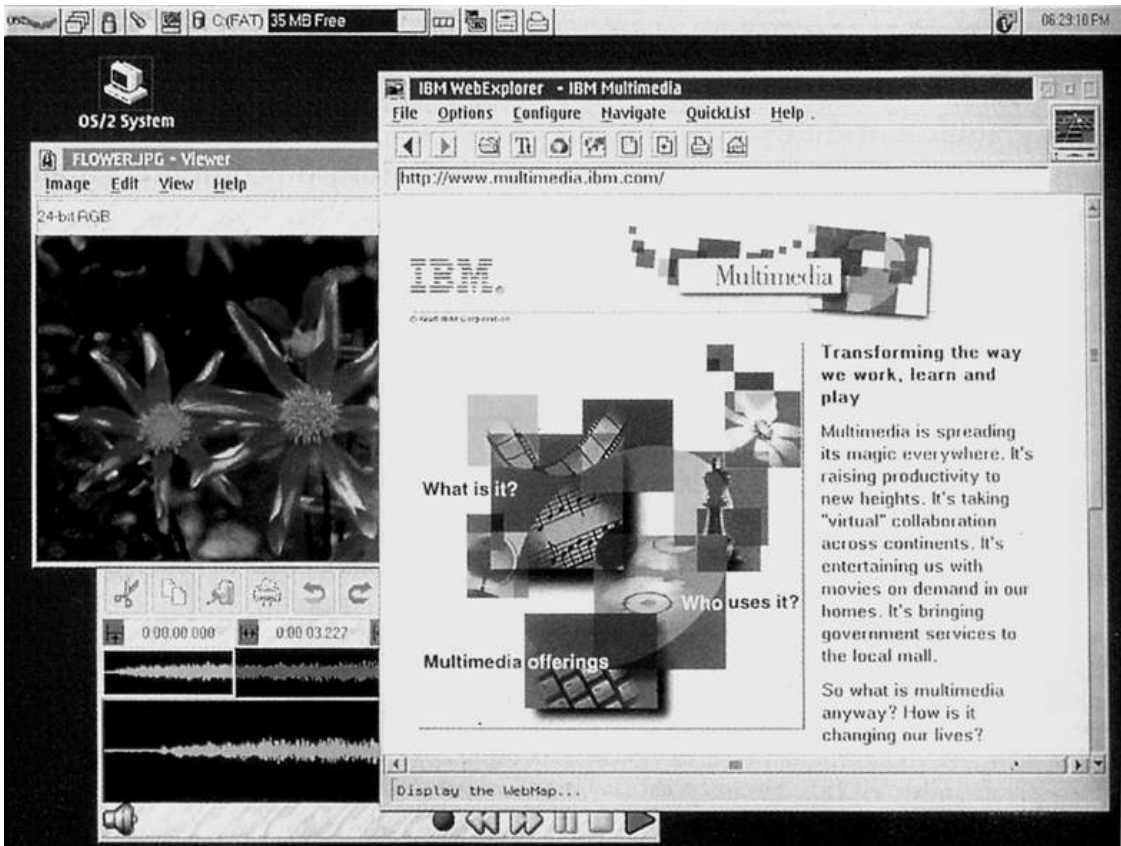
Part One	Computer Basics	1
Unit 1:	My Computer (我的电脑)	3
Section A	Computer Overview	3
Section B	Computing on the Go	13
Section C	PDA Prizefight: Palm vs. Pocket PC	19
Unit 2:	Computer Architecture (计算机硬件结构)	30
Section A	Computer Hardware	30
Section B	The Universal Serial Bus	37
Section C	DVD	43
Unit 3:	Operating System (操作系统)	48
Section A	Operating System	48
Section B	Longhorn: The Next Version of Windows	57
Section C	What Is Linux	65
Unit 4:	Computer Language and Programming (计算机语言与编程)	69
Section A	Programming Language	69
Section B	The Java Language	77
Section C	Arrays	84
Part Two	Computer Software	91
Unit 5:	Software Development (软件开发)	93
Section A	Computer Program	93
Section B	Microsoft .NET vs. J2EE	103
Section C	Distributed Systems	111
Unit 6:	Software Engineering (软件工程)	119
Section A	Software Life Cycle	119
Section B	Model Driven Development	126
Section C	What Is a Design Pattern?	134
Unit 7:	Database (数据库)	140
Section A	Enter the World of Relational Databases	140
Section B	Database Objects of SQL Server	149
Section C	.NET Data Access Architecture	154
Part Three	Network and Communication	165
Unit 8:	Telecommunications and Information Superhighway (电信与信息高速公路)	167
Section A	Telecommunications and Computer	167
Section B	Internet and Information Superhighway	175
Section C	ATM and Its Advantages	182
Unit 9:	Computer Networks (计算机网络)	190
Section A	Computer Networks	190

Section B	TCP/IP: the Basics	198			
Section C	Network Connecting Devices.....	205			
Unit 10: Internet	(因特网)	211			
Section A	How Does the Internet Work.....	211			
Section B	What's E-Mail.....	219			
Section C	Videoconferencing	222			
Part Four	IT and Our Life	229			
Unit 11: IT and Computer Application	(IT 与计算机应用)	231			
Section A	Information Revolution	231			
Section B	Geographic Information System.....	240			
Section C	Agent Technology.....	246			
Unit 12: Electronic Commerce	(电子商务)	251			
Section A	Electronic Commerce	251			
Section B	The Evolution of e-Business.....	259			
Section C	Setting Goals for Your Online Business.....	265			
Unit 13: Computer Security	(计算机安全)	270			
Section A	Computer Security	270			
Section B	Computer Virus.....	279			
Section C	Through a Hacker's Eyes.....	286			
Unit 14: Remarks by IT Giants	(IT 精英如是说)	292			
Section A	Remarks by Bill Gates.....	292			
Section B	Louis V. Gerstner Jr.: IT and the Reformation of Chinese State-Owned Enterprises	299			
Section C	Andrew S. Grove: Only the Paranoid Survive.....	306			
Part Five	Appendix	311			
Appendix A	Glossary	313			
A	313	I	325	Q	335
B	314	J	327	R	335
C	316	K	328	S	336
D	319	L	328	T	340
E	321	M	329	U	341
F	322	N	330	V	342
G	324	O	331	W	343
H	324	P	332		
Appendix B	Abbreviation				344

Unit 3: Operating System (操作系统)

Section A

Operating System



I. Introduction

An operating system (OS), in computer science, is the basic software that controls a computer. It has three major functions: it coordinates and manipulates computer hardware, such as computer memory, printers, disks, keyboard, mouse, and monitor; it organizes files on a variety of storage media, such as floppy disk, hard drive, **compact disc**, and tape; and it

compact disc
光盘

manages hardware errors and the loss of data.

II. How an Os Works

Operating systems control different computer processes, such as running a spreadsheet program or accessing information from the computer's memory. One important process is the interpretation of commands that allow the user to communicate with the computer. Some command **interpreters** are text-**oriented**, requiring commands to be typed in. Other command interpreters are graphically oriented and let the user communicate by pointing and clicking on an icon, an on-screen picture that represents a specific command. Beginners generally find graphically oriented interpreters easier to use, but many experienced computer users prefer text-oriented command interpreters because they are more powerful.

Operating systems are either single-tasking or **multitasking**. The more primitive single-tasking operating systems can run only one process at a time. For instance, when the computer is printing a document, it cannot start another process or respond to new commands until the printing is completed.

All modern operating systems are multitasking and can run several processes simultaneously. In most computers there is only one central processing unit (CPU), (the computational and control unit of the computer), so a multitasking OS creates the illusion of several processes running simultaneously on the CPU. The most common mechanism used to create this illusion is **time-slice** multitasking, **whereby** each process is run individually for a fixed period of time. If the process is not completed within the **allotted** time, it is suspended and another process is run. This exchanging of processes is called **context switching**. The OS performs the "**bookkeeping**" that preserves the state of a suspended process. It also has a mechanism, called a **scheduler**, that determines which process will be run next. The scheduler runs short processes quickly to **minimize perceptible** delay. The processes appear to run simultaneously because the user's sense of time is much slower than the processing speed of the computer.

Operating systems can use **virtual memory** to run processes that require more **main memory** than is actually available. With this technique, space on the hard drive is used to mimic the extra memory needed. Accessing the hard drive is more time-consuming than accessing main memory, however, so performance of the computer slows.

interpreter

/in'tə:prɪtə/

n. 解释程序,
解释器

orient

'ɔ:riənt/
v. 给...定向; 使适应; 使面向

multitasking

/'mʌlti,tə:skɪŋ/

n. 多任务处理, 同时执行多项任务

time-slice

/'taɪmsleɪs/
n. 时间片

whereby

'weə'baɪ/
ad. 靠那个; 借以

allot

'ə'lɒt/
n. 分配; 分派

context switching

上下文转换

bookkeeping

/'bʊk,ki:piŋ/
n. 簿记, 登录账目

scheduler

/'ʃedju:ələ; 'ske-
dʒu:ələ/
n. 调度程序, 计划任务程序

minimize

/'mɪnɪmaɪz/
v. 使减少到最低限度

perceptible

/'pe'septəbl/
a. 可感知的; 可察觉的

virtual memory

虚拟内存

main memory

主存 (存储器)

III. Current Operating Systems

The operating systems commonly found on personal computers are UNIX, Macintosh OS, MS-DOS, OS/2, and Windows.

1. UNIX

UNIX, in computer science, is a multiuser, multitasking operating system originally developed by Ken Thompson¹ and Dennis Ritchie² at AT&T³ Bell Laboratories⁴ in 1969 for use on minicomputers. UNIX exists in various forms and implementations; among these are versions developed at the University of California at Berkeley⁵ (known as BSD⁶ **releases**) and versions released by AT&T, the most recent being AT&T System V. UNIX is considered a powerful operating system which, because it is written in the C language, is more portable—less machine-specific—than other operating systems. UNIX is available in several related forms, including AIX, a version of UNIX adapted by IBM (to run on RISC⁷-based workstations), A/UX (a graphical version for the Apple⁸ Macintosh), and Mach (a rewritten but essentially UNIX-**compatible** operating system for the NeXT computer).

Its popularity is due in large part to the growth of the interconnected computer network known as the Internet, the software for which initially was designed for computers that ran UNIX. Variations of UNIX include SunOS (distributed by SUN Microsystems Inc.⁹), Xenix (distributed by Microsoft Corporation), and Linux¹⁰. UNIX and its clones support multitasking and multiple users. Its file system provides a simple means of organizing disk files and lets users protect their files from other users. The commands in UNIX are not intuitive, however, and mastering the system is difficult.

release / ri'li:s /
n. 版本; 发布

compatible
/ kəm'pætəbl /
a. 兼容的

-
- ¹ *Ken Thompson*: 肯·汤普森 (1943~), 美国计算机科学家, 与丹尼斯·里奇一起开发了 UNIX 操作系统。
² *Dennis Ritchie*: 丹尼斯·里奇 (1941~), 美国计算机科学家, 开发了 C 语言, 并与肯·汤普森一起开发了 UNIX 操作系统。
³ *AT&T*: 美国电话电报公司 (American Telephone and Telegraph 的缩略)。
⁴ *Bell Laboratories*: 贝尔实验室。
⁵ *University of California at Berkeley*: 加利福尼亚大学伯克利分校。
⁶ *BSD*: 伯克利软件发行中心 (*Berkeley Software Distribution* 的首字母缩略)。
⁷ *RISC*: 精简指令集计算机 (*reduced instruction set computing* 的首字母缩略)。
⁸ *Apple*: (美国) 苹果计算机公司。
⁹ *SUN Microsystems Inc.*: (美国) 太阳微系统股份有限公司, 主要生产 SUN 系列工作站和网络产品, 是 JAVA 语言的创始者。SUN 系 *Stanford University Network* 的首字母缩略。
¹⁰ *Linux*: Linux 操作系统, 自 UNIX 发展而来, 由芬兰人 Linus Torvalds 于 1991 年编写成功, 随后采用开放源代码方式在 Internet 上由全球编程人员共同开发, 获得极大成功。

2. MS-DOS

oversee /,əʊvə'si:/
v. 监视; 检查;
管理

MS-DOS is the acronym for Microsoft Disk Operating System. In computer science, MS-DOS—like other operating systems—**oversees** such operations as disk input and output, video support, keyboard control, and many internal functions related to program execution and file maintenance. MS-DOS is a single-tasking, single-user operating system with a **command-line** interface.

command line
命令行

3. OS/2

OS/2, or Operating System 2, is an operating system developed for the personal computer in the mid-1980s by International Business Machines Corporation (IBM) and Microsoft Corporation. At the time OS/2 was introduced in late 1987, the most common personal computers were IBM-compatible computers running the Microsoft Disk Operating System (MS-DOS) and computers manufactured by Apple Computer Corporation running Apple's system for the Macintosh (Mac OS). The Macintosh operating system included multitasking, a feature that enabled computers to run several applications simultaneously. In a computer network, multitasking allows several users on different computers to have simultaneous access to the same application or **data set**. OS/2 was the first operating system designed for IBM-compatible personal computers that allowed multitasking.

data set
数据集

graphical user interface
图形用户界面
presentation manager
表示管理程序

The first version of OS/2, version 1.0, was text-oriented and lacked a **graphical user interface** (GUI) that would allow users to enter commands with a point-and-click input device, such as a computer mouse. A year later IBM and Microsoft released OS/2 version 1.1, which included a GUI called the **Presentation Manager**. The Presentation Manager interface contained icons, pictures or words on the screen that users could click on with a mouse to enter instructions. OS/2 version 1.1 also allowed users to have multiple windows open (windows are portions of the screen that each contain a different document or program) and included **pull-down** lists of commands that the user could choose by clicking on them with their mouse.

pull-down
/'pʊldaʊn/
a. 下拉(式)的
collaboration
/kə,læbə'reiʃən/
n. 合作, 协作

IBM and Microsoft ended their **collaboration** on OS/2 in 1991 after Microsoft released its Windows software, a multitasking environment that ran on MS-DOS. In 1992 IBM released version 2.0 of OS/2, which ran Microsoft Windows programs and could perform multitasking of DOS operations. It also contained an object-oriented programming environment

that allowed software designers to create programs using high-level, object-oriented programming languages.

Subsequent versions of OS/2 offered enhanced performance and multimedia capabilities, and in 1994 IBM announced that more than 5 million copies of OS/2 had been sold since its introduction. The same year, IBM introduced a new version of OS/2 called OS/2 Warp that featured improved performance, more multimedia capabilities, an **array** of integrated applications, and easy access to the Internet. IBM has continued to **upgrade** and extend OS/2 Warp.

array /ə'rei/
n. 一系列, 大量
upgrade /ʌp'greɪd/
v. 使升级; 改善

4. Windows

Windows, in computer science, is a personal computer operating system sold by Microsoft Corporation that allows users to enter commands with a point-and-click device, such as a mouse, instead of a keyboard. The Windows operating system provides users with a graphical user interface (GUI), which allows them to manipulate small pictures, called icons, on the computer screen to issue commands. Windows is the most widely used operating system in the world. It is an extension of and replacement for Microsoft's Disk Operating System (MS-DOS).

The Windows GUI is designed to be a natural, or intuitive, work environment for the user. With Windows, the user can move a cursor around on the computer screen with a mouse. By pointing the cursor at icons and clicking buttons on the mouse, the user can issue commands to the computer to perform an action, such as starting a program, accessing a data file, or copying a data file. Other commands can be reached through pull-down or **click-on** menu items. The computer displays the active area in which the user is working as a window on the computer screen. The currently active window may **overlap** with other previously active windows that remain open on the screen. This type of GUI is said to include WIMP¹ features: windows, icons, menus, and pointing device (such as a mouse).

click-on /'klikɒn/
a. (鼠标) 点击
(式) 的
overlap /,əʊvə'læp/
v. 重叠, 层叠

innovation
/,ɪnə'veɪʃən/
n. 革新, 创新

Computer scientists at the Xerox Corporation's² Palo Alto Research Center (PARC³) invented the GUI concept in the early 1970s, but this **innovation** was not an immediate commercial success. In 1983 Apple Computer featured a GUI in its Lisa computer. This GUI was updated and

¹ **WIMP**: 窗口-图标-菜单-指点设备 (windows-icons-menus-pointing device 的首字母缩略)。

² **Xerox Corporation**: 施乐公司。

³ **PARC**: (施乐公司) 帕洛阿尔托研究中心 (Palo Alto Research Center 的首字母缩略)。

improved in its Macintosh computer, introduced in 1984.

tiled / taɪld /
a. 平铺（式）的

Microsoft began its development of a GUI in 1983 as an extension of its MS-DOS operating system. Microsoft's Windows version 1.0 first appeared in 1985. In this version, the windows were **tiled**, or presented next to each other rather than overlapping. Windows version 2.0, introduced in 1987, was designed to resemble IBM's OS/2 Presentation Manager, another GUI operating system. Windows version 2.0 included the overlapping window feature. The more powerful version 3.0 of Windows, introduced in 1990, and subsequent versions 3.1 and 3.11 rapidly made Windows the market leader in operating systems for personal computers, in part because it was prepackaged on new personal computers. It also became the favored platform for software development.

network / 'netwɜ:k /
v. 联网

In 1993 Microsoft introduced Windows NT (New Technology). The Windows NT operating system offers 32-bit multitasking, which gives a computer the ability to run several programs simultaneously, or in parallel, at high speed. This operating system competes with IBM's OS/2 as a platform for the intensive, high-end, **networked** computing environments found in many businesses.

sleek / sli:k /
a. 光滑的；线条明快的，造型优美的

In 1995 Microsoft released a new version of Windows for personal computers called Windows 95. Windows 95 has a **sleeker** and simpler GUI than previous versions. It offers 32-bit processing, efficient multitasking, network connections, and Internet access.

window / 'wɪndəu /
v. 把…分隔成多个窗口

Few computer users run MS-DOS or OS/2 directly. They prefer versions of UNIX or **windowing** systems with graphical interfaces, such as Windows 98 or the Macintosh OS, which make computer technology more accessible. However, graphical systems generally have the disadvantage of requiring more hardware—such as faster CPUs, more memory, and higher-quality monitors—than command-oriented operating systems.

IV. Future Technologies

Operating systems continue to evolve. A recently developed type of OS called a distributed operating system is designed for a connected, but independent, collection of computers that share resources such as hard drives. In a distributed OS, a process can run on any computer in the network (presumably a computer that is idle) to increase that process's performance. All basic OS functions—such as maintaining file systems, ensuring reasonable behavior, and recovering data in the event of a partial

failure—become more complex in distributed systems.

Research is also being conducted that would replace the keyboard with a means of using voice or handwriting for input. Currently these types of input are imprecise because people pronounce and write words very differently, making it difficult for a computer to recognize the same input from different users. However, advances in this field have led to systems that can recognize a small number of words spoken by a variety of people. In addition, software has been developed that can be taught to recognize an individual's handwriting.

Exercises

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

1. An operating system is the basic _____ that controls a computer.
2. If you want to run more than one process at a time, you must use an operating system with _____ capability.
3. Operating systems can use _____ memory to run processes that require more main memory than is actually available.
4. OS/2 is an operating system developed jointly by _____ and Microsoft Corporation for the personal computer.
5. _____, whose popularity is due in large part to the growth of the Internet, is a(n) _____, multitasking operating system.
6. The most widely used operating system in the world is _____, which provides users with a(n) _____ user interface.
7. MS-DOS, the acronym for Microsoft Disk Operating System, is a single-tasking, single-user operating system with a(n) _____ interface.
8. A(n) _____ OS is designed for a connected, but independent, collection of computers that share resources.

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

- | | |
|--------------------------------|------------|
| 1. data set | 11. 命令行界面 |
| 2. pointing device | 12. 多任务化环境 |
| 3. graphical user interface | 13. 电子制表程序 |
| 4. time-slice multitasking | 14. 主存 |
| 5. object-oriented programming | 15. 存储介质 |
| 6. click on an icon | 16. 磁盘文件 |
| 7. context switching | 17. 命令解释器 |
| 8. distributed system | 18. 网络连接 |
| 9. pull-down lists of commands | 19. 磁盘操作系统 |
| 10. simultaneous access | 20. 拷贝数据文件 |

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

<i>program</i>	<i>interface</i>	<i>system</i>	<i>user</i>
<i>storage</i>	<i>classify</i>	<i>unauthorized</i>	<i>control</i>
<i>document</i>	<i>efficiently</i>	<i>function</i>	<i>detect</i>
<i>internal</i>	<i>security</i>	<i>password</i>	<i>input</i>

Operating systems for micro, mini, and mainframe computers perform many services. These services can be _____ either as “external” or “internal.”

The operating system provides external services that help _____ start programs, manage stored data, and maintain _____. You, as the computer user, control these external _____. Using a command-line, menu-driven, or GUI user _____, an operating system provides you with a way to select the _____ you would like to use. The operating system also helps you find, rename, and delete _____ and other data stored on disk or tape. On many, but not all computer _____, the operating system helps you maintain security by checking your *user ID* (用户标识) and _____, as well as protecting your data from _____ access and *revisions* (修改).

The operating system provides _____ services “behind the scenes” to ensure that the computer system functions _____. These internal services are not generally under your _____, but instead are controlled by the operating system itself. The operating system controls _____ and output, *allocates* (分配) system resources, manages the _____ space for programs and data, and _____ equipment failure without any direction

from you.

IV. Translate the following passage from English into Chinese:

Multitasking, in computer science, is a mode of operation offered by an operating system in which a computer works on more than one task at a time. There are several types of multitasking. Context switching is a very simple type of multitasking in which two or more applications are loaded at the same time but only the *foreground* (前台的) application is given processing time; to *activate* (激活) a *background* (后台的) task, the user must bring the window or screen containing that application to the front. In *cooperative* (合作的) multitasking, background tasks are given processing time during idle times in the foreground task (such as when the application waits for a keystroke), and only if the application allows it. In time-slice multitasking each task is given the microprocessor's attention for a fraction of a second. To maintain order, tasks are either assigned priority levels or processed in *sequential* (顺序的) order. Because the user's sense of time is much slower than the processing speed of the computer, time-slice multitasking operations seem to be simultaneous.

Section B

Longhorn¹: The Next Version of Windows



code-name

/kəʊd'neim/

v. 给...规定代号

embryonic

/ˌembri'ɒnik/

a. 萌芽期的, 初期的

divulge

/daɪ'vʌldʒ/

v. 泄露 (秘密等)

roundup

/'raʊndʌp/

n. 聚拢; 聚会

By all accounts², the next version of Windows will be a long time coming. **Code-named** Longhorn, the OS isn't due until sometime in 2006. Bits and pieces³ of the **embryonic** operating system have been leaking onto the Internet since the beginning of the year, however, and recently, Microsoft has started to **divulge** details of its "biggest bet" since Windows 95.

At a **roundup** in Los Angeles, the company handed out DVDs with an

¹ *Longhorn*: 长角, 微软使用的开发代号。它和 Windows XP 的开发代号 Whistler 一样, 是微软总部附近滑雪胜地的地名。Longhorn 位于 Whistler 山脚下, 是一处为前来滑雪的人们提供服务的场所。

² *by all accounts*: 根据各种流传的说法, 根据许多人所说。

³ *bits and pieces*: 零星 (碎片), 零散。

corral /kɔ:'rɑ:l/

v. 把...集合在一起, 搜集

application

/,æpli'keiʃən/ n. 应用程序, 应用软件

safeguard

/'seifgɑ:d/

v. 保护, 维护

terabyte /'terəbait/

n. 太字节, 兆兆字节

graphics

accelerator

图形加速器

animation

/,æni'meiʃən/ n. 动画(制作); 直观显示

intact /in'tækt/

a. 完整无缺的

tile /tail/

n. 瓦片; 平铺显示

active desktop

活动桌面

early version of Longhorn, and it detailed changes to the OS along with related programming tools such as Visual Studio¹ and SQL Server². The goal was to **corral** developers behind Longhorn, but in the process, Microsoft also gave a glimpse at what's in store for the rest of us.

While it will be compatible with today's **applications**, Longhorn will change everything about the OS, from the way the desktop looks to how you store and retrieve files to how your PC is **safeguarded**. And it should provide the foundation for more-innovative Windows applications.

I. Looks Are Everything

By the time Longhorn ships, according to Microsoft chairman Bill Gates, PCs will have 4GHz³ to 6GHz processors, more than 2GB of memory, at least a **terabyte** of storage, and **graphics accelerators** three times more powerful than those offered by ATI⁴ and NVIDIA⁵ today. He says that Longhorn is designed to take advantage of all this muscle⁶, and nowhere is that more evident than in the rich, three-dimensional interface known as Aero⁷.

The Aero eye candy⁸ is built on a new graphics and presentation engine (Avalon⁹), which in turn is based on XML¹⁰ but also borrows from Microsoft technologies such as DirectX¹¹ and ClearType¹². The result is a more sophisticated look with transparent menus, smooth **animation**, and windows that shrink to thumbnail size or expand with the contents **intact**.

The new engine also lets you create and display **Tiles**¹³ on the desktop. Conceptually, these remind us of the old **Active Desktop**. The difference

¹ *Visual Studio*: 可视化工作室, 微软的软件开发工具套件。

² *SQL*: 结构化查询语言 (structured query language 的首字母缩略)。SQL Server 系微软的关系型数据库管理系统。

³ *GHz*: 千兆赫 (gigahertz 的缩略)。

⁴ *ATI*: 著名的显示芯片生产商。

⁵ *NVIDIA*: 著名的显示芯片生产商。

⁶ *muscle*: 在这里指“基本或本质的东西”、“主要部分”。

⁷ *Aero*: 字面意思是“飞行”, 这里被微软用作 Longhorn 用户界面的名称。Aero 整个界面的画质饱满, 而且富有弹性和动感。无论是静态图像, 还是动态动画, 都有很高的水准。

⁸ *eye candy*: 指 someone or something superficially attractive to look at, 即看起来很舒服、很顺眼的人或物。

⁹ *Avalon*: 阿瓦隆。凯尔特族传说中的西方乐土岛, 据说亚瑟王及其部下死后尸体被移往该岛。这里指微软公司开发的新图形与显示引擎的代号。

¹⁰ *XML*: 可扩展标记语言 (Extensible Markup Language 的缩略)。

¹¹ *DirectX*: 微软公司的对硬件编程的接口, 包括 DirectDraw、DirectSound 等。

¹² *ClearType*: 微软公司对字体显示进行优化的一种技术。

¹³ *Tiles*: Longhorn 中使用的一种信息区块, 参见插图 Figure 3B-1。Tiles 来自 tile, 字面意思为“瓦片”, 在计算机用语中表示“平铺显示”。

online /'ɒn'lain/
a. & ad. 联机的
(地), 在线的
(地)
stock quote
股票行情

this time, however, is that instead of HTML, Tiles are based on XML, so they can display both **online** information, such as news headlines or **stock quotes**, and info from any local drive, such as all your MP3s or a list of

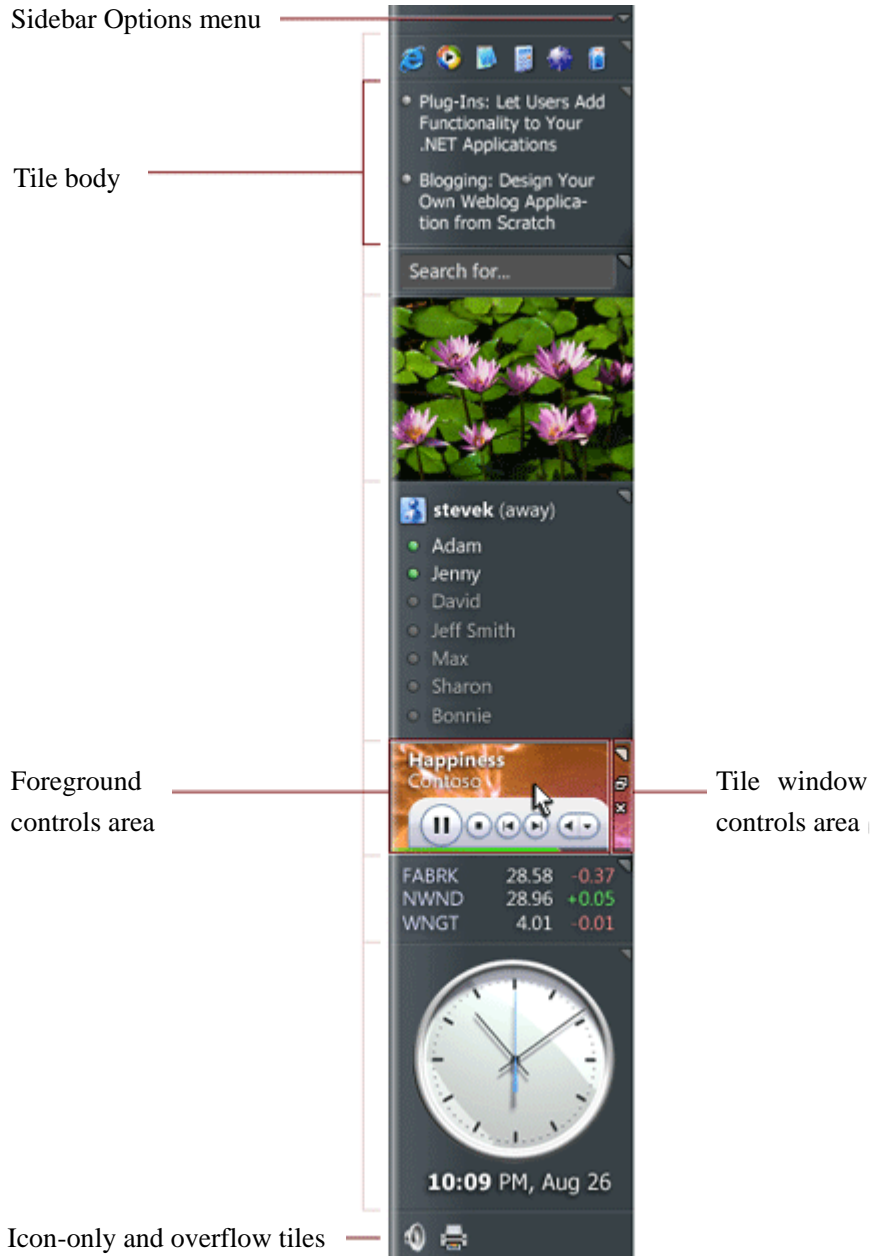


Figure 3B-1: An example of the sidebar with tiles

sidebar /'saɪdba:/
n. 工具边条

important contacts. The **Sidebar**¹—one of the prominent new features of

¹ **Sidebar:** Longhorn 中使用的一种工具条框, 参见插图插图 Figure 3B-1 和 Figure 3B-2。

buddy /'bʌdi/
n. 伙伴, 好朋友
control /kən'trəʊl/
n. 控件, 控制器
quick-launch key
快速加载键

Longhorn—is really just a collection of Tiles showing a clock, a **buddy** list, news, Windows Media Player **controls**, and **quick-launch keys** for major applications. Despite the name, it can be positioned vertically on the side of your display or horizontally across the top or the bottom. Early versions have also reportedly included Tiles for synchronizing with other devices and for searching.

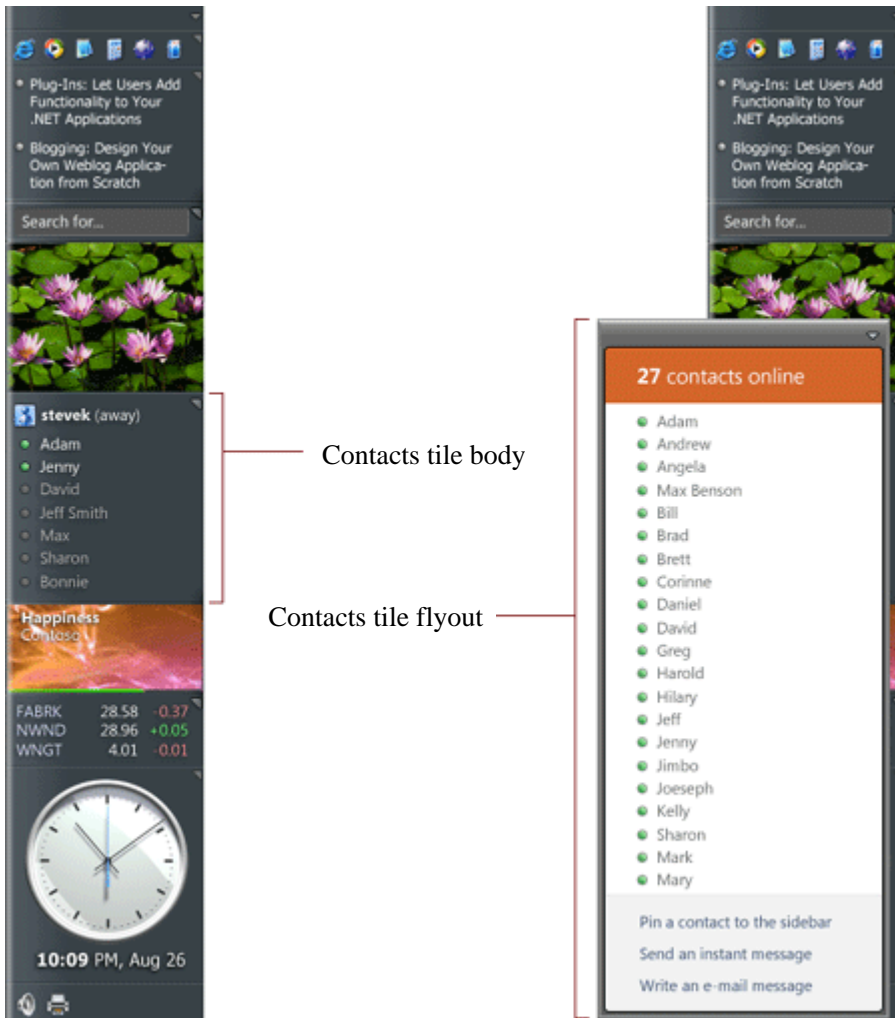


Figure 3B-2: Tile and flyout for a contact list

tablet /'tæblɪt/
n. 便笺簿; (图形) 输入板; 药片
tablet PC
便笺簿式个人电脑
Media Center PC
媒体中心个人电脑

Longhorn's user interface will integrate many of the cutting-edge features that Microsoft is testing today in **tablet PCs**, mobile devices, and **Media Center PCs**. Microsoft has shown off Longhorn's support for ink handwriting recognition and voice. Just as Internet Explorer became a

morph /mɔ:f/
v. 变化, 改变
omnipresent
/ˌɒnɪˈprezənt/
a. 无所不在的, 普遍存在的
standalone
/ˈstændəˌleɪn/
a. 独立的

sexy /ˈseksi/
a. 性感的; 富有魅力的, 迷人的
hype /haɪp/
n. 天花乱坠的广告宣传; 大肆宣传
warrant /ˈwɒrənt/
v. 证明...是正当(或有理)的
relational database
关系型数据库
coincidence
/kəʊˈɪnsɪdəns/
n. 巧合; 巧事
stash /stæʃ/
v. <口> 存放, 贮藏
lurk /lə:k/
v. 潜伏; 暗藏; 潜行

silos /ˈsaɪləʊ/
n. 筒仓(储存粮食、水泥等散货的立式圆柱形仓库)
reside /ˈrɪzaɪd/
v. 居住; 存在
holy grail /greɪl/
[通常作 **H-G**](传说耶稣最后晚餐用的)圣杯, 圣盘

component of Windows years ago, Windows Media Player is slowly **morphing** into an **omnipresent** feature rather than a **standalone** application, and some versions of Longhorn even include a My TV application.

This requires not only a fast PC, but a relatively large high-resolution display. Microsoft is gambling that most users will have both by the time Longhorn rolls around¹. But just in case, the OS will reportedly offer different levels of operation that disable the more intensive Aero effects to boost performance on less capable PCs.

II. Minding the Store

Who knew storage could be **sexy**? The most talked-about feature in Longhorn so far is its new storage system, WinFS². In this case, the **hype** may be **warranted**. WinFS is a file system that works more like a **relational database**, which is no **coincidence**, since it is based on the same technology as the next version of SQL Server, code-named Yukon. Rather than **stashing** data in a series of folders and subfolders, WinFS promises to let you organize and search for files based on their content or other attributes.

The old folders and subfolders will still **lurk** beneath the surface, but files can be organized into stacks. Though these stacks have been described as an extension of the existing My Pictures or My Music folders, they appear to work more like the search folders in Outlook 2003. In other words, the actual location of the files isn't really important. Instead, these virtual folders show you files or data that all share common properties using filters that you can create and modify. For example, you could create stacks that show all songs longer than one minute, display photos taken in the past month, or sort all contacts (Longhorn includes a My Contacts folder) by e-mail address.

Microsoft also promises that WinFS will make searching for files much more powerful by breaking down the separate **silos** of information that today **reside** in application-specific file formats. For example, from a single search box, you will be able to pull up all e-mail messages, Web pages, and Office documents related to a particular topic—a powerful feature that Gates has referred to as his personal **holy grail**.

¹ *by the time Longhorn rolls around*: 到 Longhorn 投入使用时。

² *WinFS*: 微软新一代操作系统所使用的文件系统。

batten / 'bætən /
v. 用板条(或压条等)使固定

hatch / hætf /
n. 舱口(盖)

corporate
/ 'kɔ:pəret /
a. 公司的; 社团的

draconian
/ drə'keunian /
a. 严厉的; 严酷的

mull / mʌl /
v. <口> 仔细考虑

built-in / 'bilt'in /
a. 内置的, 内建的, 嵌入的, 内在的

firewall / 'faɪəwɔ:l /
n. 防火墙

default / di'fɔ:lt /
n. 默认, 缺省, 系统设定值

swirl / swə:l /
v. 纷乱地流传

stringent
/ 'strɪndʒənt /
a. 严格的; 严厉的

authentication
/ ɔ:θenti'keɪʃən /
n. 验证, 认证

encryption
/ in'kriptʃən / n. 加密

chipset / 'tʃɪpsət /
n. 芯片组

hood / hud /
n. 风帽; 罩; 车盖

III. Batten Down the Hatches¹

Microsoft's Achilles' heel² has been security. (The company recently admitted that security fears were affecting its **corporate** sales.) With that in mind, a major focus of Longhorn will be on improving security, though exactly how Microsoft plans to do so without imposing **draconian** measures on users remains unclear.

Some changes may be apparent to users even now. For example, Microsoft is **mulling** whether to turn on the **built-in firewall** by **default**, and rumors are **swirling** that Longhorn will include its own antivirus program. It seems more likely, however, that the company will find a way to integrate existing antivirus utilities deeper into the OS to push more users to install and update them regularly. Early versions have also included a tool that lets parents restrict access to the PC or Internet to certain times of the day. A new reporting tool, basically a "black box"³ for your PC, could also help developers locate and plug security holes in applications.

On a broader level, Microsoft is developing a new security architecture, called NGSCB⁴ (Next-Generation Secure Computing Base), that splits the OS into two parts: a standard mode, where today's hardware and software play, and a secure mode that requires more **stringent authentication** and **encryption**. The move requires changes to third-party hardware and software as well. For example, Intel has stated that its next-generation Prescott chips and **chipsets**⁵, due out early next year⁶, will support it.

IV. Under the Hood

Ordinarily we wouldn't spill much ink on changes to Windows at the programming level, but Microsoft claims that these changes will be the foundation for more-sophisticated applications.

At the heart of Longhorn is a shift toward XML. Microsoft describes

¹ *Batten down the hatches*: 封住舱口, 这里指补上安全漏洞。

² *Achilles' heel*: 阿喀琉斯的脚踵, 致命弱点, 唯一的弱点。根据希腊神话, 阿喀琉斯(或译阿基里斯)出生后被其母亲握脚踵倒提着在冥河水中浸过, 因此除未浸到水的脚踵外, 浑身刀枪不入。

³ *black box*: 黑盒子, 黑匣子。black box 是装在飞机上的一种自动电子记录仪器, 用以收集并记录飞行资料, 常用来判定飞机失事的原因。这里指 Longhorn 提供的一种可帮助开发人员找到和弥补应用程序中安全漏洞的报告工具, 因为在功能上类似于飞机上的黑盒子, 故作此比喻。

⁴ *NGSCB*: 下一代安全计算基础(*Next-Generation Secure Computing Base* 的首字母缩略)。

⁵ *Prescott chips and chipsets*: Prescott (普雷斯科特) 芯片和芯片组。Prescott 是英特尔公司新推出的下一代 Pentium 4 处理器。

⁶ *due out early next year*: 按预定计划明年初推出。

schema /'ski:mə/
n. 模式; 图表; 纲要

geek /gi:k/
n. 怪杰, 奇才; 怪人; 呆子

-speak /spi:k/
comb. form 表示“行话”, “语言”

messaging
/'mesidʒɪŋ/
n. 通信, 消息接发

suite /swi:t/
n. (软件的) 套件; (同类物的) 系列, 套

gamble /'gæmbəl/
n. 赌博; 投机, 冒险

waiting game
伺机而动的策略

hazy /'heizi/
a. 雾蒙蒙的; 模糊的, 不明确的

bottom line
账簿底线; 基本意思 (或情况)

deliver /dɪ'lɪvə/
v. 履行; 实现 (*on*)

upgrade /'ʌpgeɪd/
n. 升级, 提高品级

this as moving “those rich user **schemas** down into the platform,”¹ which is **geek-speak** for taking all that data stuck in proprietary application file formats and putting it in XML files that can be shared. The obvious example is a common address book that can be used by any PIM², e-mail program, instant-**messaging** client, and so on. Similarly, the new WinFX programming interface, along with tools such as the Indigo³ communications platform, will standardize how applications interact with that XML data.

Microsoft believes so strongly in the power behind Longhorn that it is putting most of its Office resources behind creating a version of the **suite** that runs solely on the new OS—a major **gamble**. The company also plans to release new versions of Visual Studio for developing Longhorn applications.

V. Waiting Game

Before you get in line at CompUSA⁴, keep in mind that Microsoft is still very **hazy** on the release dates for Longhorn. Most seem to agree it will be 2006 before it hits stores⁵, but the company says only that it will be a technology-driven release, which is a nice way of saying that the date will probably slip. A beta version of the client OS, however, is expected sometime next summer. The cost will be about the same as Windows XP, according to Gates. The **bottom line**: If Microsoft **delivers** on half of what's supposed to be in Longhorn, it will still be a significant **upgrade**.

Exercises

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

1. NGSCB, the new security architecture Microsoft is developing for Longhorn, splits the OS into two parts: a standard mode and a(n) _____ mode.
2. It is reported that Longhorn will provide different levels of

¹ Microsoft describes this as moving “those rich user schemas down into the platform”: 微软形容这就相当于把那些多种多样的用户模式迁移到这个 (使用 XML 的) 平台上。(后面句子说明这么做是为了共享数据。)

² PIM: 个人信息管理程序 (personal information manager 的首字母缩略)。

³ Indigo: Longhorn 中的通信平台。

⁴ CompUSA: 北美最大的计算机零售商。

⁵ before it hits stores: 在它摆在商店里销售之前。在这里, hit 是一种口语用法, 表示“出现在”。

operation that disable the more intensive Aero effects to boost _____ on less capable PCs.

3. With Longhorn's new graphics and presentation engine, we can create and display Tiles on the desktop, which remind us of the old Active Desktop but are based on _____ instead of _____.

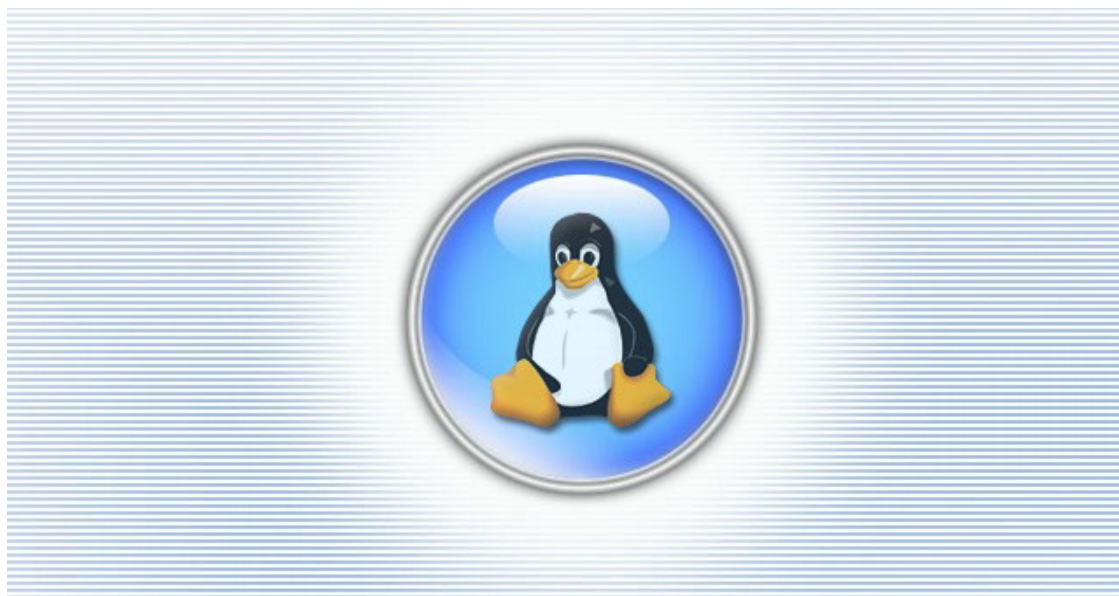
4. The most talked-about feature in Longhorn so far is its new storage system, WinFS, which works like a(n) _____ database.

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

- | | |
|---------------------------|----------|
| 1. search box | 6. 专有软件 |
| 2. built-in firewall | 7. 快速加载键 |
| 3. standalone application | 8. 图形加速器 |
| 4. active desktop | 9. 虚拟文件夹 |
| 5. mobile device | 10. 三维界面 |

Section C

What Is Linux



scratch /skrætʃ/

n. 起跑线; 零分

compliance

/kəm'plaiəns/

n. 遵从; 顺从

driver /'draɪvə/

n. 驱动器, 驱动程序

library /'laɪbrəri/

n. 程序库, 库

port /pɔ:t/

n. 端口, 通信口

fledged /fledʒd/

a. 羽毛已长成的;
成熟的

Linux is a free UNIX clone written from **scratch**¹ by Linus Torvalds² and a team of programmers over the Internet. Linux aims towards POSIX³ **compliance** (a set of standards that show what a UNIX should be).

It contains all of the features that you would expect in not only a UNIX, but any operating system. Some of the features included are true multitasking, virtual memory, the world's fastest TCP/IP⁴ **drivers**, shared **libraries**, and of course multi-user capabilities (this means hundreds of people can use one computer at the same time, either over a network, the Internet, or on laptops/computers or terminals connected to the serial **ports** of that computer). Linux runs fully in protected mode (unlike Windows) and supports fully **fledged** 32-bit and 64-bit multitasking.

¹ *from scratch*: 从零开始, 白手起家。

² *Linus Torvalds*: 莱纳斯·托瓦尔兹 (1969~), 芬兰人, Linux 操作系统的创始人。

³ *POSIX*: 可移植的 UNIX 操作系统 (*Portable Operating System Interface for UNIX* 的首字母或尾字母缩略)。

⁴ *TCP/IP*: TCP/IP 协议, 传输控制协议 / 网际协议 (*Transmission Control Protocol/Internet Protocol* 的首字母缩略)。

conform /kəm'fɔ:m/
v. 遵照; 符合 (*to, with*)

uninitiated
/ˌʌnɪ'nɪʃieɪtɪd/
a. 缺乏某种特定知识和经验的

installer /in'stɔ:lə/
n. 安装程序

emulator
/'emjuleɪtə/
n. 仿真器;
仿真程序

buffer /'bʌfə/
v. 缓冲, 缓存

collaborate
/kə'læbəreɪt/
v. 合作, 协作
packet radio
无线电包通信, 无线
电分组通信
protocol
/'prəʊtəkɒl/
n. 协议

Linux also has a completely free X Windows¹ implementation **conforming** to the X/Open² standard. Most existing X Based³ programs will run under Linux without any modification. X Windows for the **uninitiated**⁴ is a GUI, similar to Microsoft Windows but is feature packed (and is rather large, consuming around 15-20MB).⁵ Most Linux distributions come completely pre-configured to a factory configuration, and distributions such as RedHat Linux also have graphical based configuration utilities and **installers** (not unlike Windows' Control Panel).

In addition, programs intended for SCO Unix 4.2 and SVR4 will run on most Linux systems unaltered (this is due to the assistance of a driver called IBCS), for example Corel Draw!⁶ for SCO and the Dataflex database system. The Linux DOS **emulator**, DOSEMU, will run packets of MS-DOS applications, including some that require VGA⁷ or SVGA⁸ graphics capabilities. Windows programs can also be run inside of X-Windows with the help of an emulator called WINE. Usually, Windows programs can run up to 10 times faster than on a native system⁹, due to Linux's **buffering** capabilities.

As usually expected from most Unices¹⁰, Linux includes advanced networking capabilities. Since the people developing Linux **collaborated** and used the Internet for their development efforts, networking support came early in Linux's development stage. Networking support in Linux is superior to most other operating systems. Linux supports connection to the Internet or any other network using TCP/IP or IPX¹¹ via Ethernet¹², fast Ethernet¹³, ATM¹⁴, modem, **packet radio** (X.25¹ **protocol**), ISDN², **token**

¹ X Windows: 由美国麻省理工学院技术人员开发的在 UNIX 工作站上使用的图形窗口, 支持客户机/服务器模式。不同于微软的 Windows。简称 X。

² X/Open: 支持 UNIX 开放标准的厂商联盟, 也称 X/Open group。他们提供的系统在任何 UNIX 模块上通用。

³ X Based: 基于 X Windows 的。

⁴ the uninitiated: 属定冠词加形容词表示一类人或事物的结构, 指“缺乏某种特定知识和经验的人”。

⁵ X Windows for the uninitiated is a GUI, similar to Microsoft Windows but is feature packed (and is rather large, consuming around 15-20MB): 为缺乏经验者开发的 X Windows 是一种图形用户界面, 类似于微软的 Windows, 但其功能被封装起来 (而且比较庞大, 耗费大约 15~20 兆空间)。

⁶ Corel Draw!: (加拿大) Corel 公司发行的一套绘图程序。

⁷ VGA: 视频图形阵列 (Video Graphics Array 的首字母缩略)。

⁸ SVGA: 超级视频图形阵列 (Super Video Graphics Array 的首字母缩略)。

⁹ a native system: 原属系统, 这里指 Windows 系统。

¹⁰ Unices: Unix 的复数形式。

¹¹ IPX: 网间报文交换, 互连网络包交换 (Internetwork Packet Exchange 的首字母缩略)。

¹² Ethernet: 以太网 (标准)。

¹³ fast Ethernet: 快速以太网, 支持 100Mbps 容量的以太网。

¹⁴ ATM: 异步传输模式 (Asynchronous Transfer Mode 的首字母缩略)。

token ring
令牌环, 令牌环

token ring, or PLIP³ (modified printer cable to another computer). As an Internet/WWW server, Linux is a very good choice, often out performing Windows NT, Novell⁴ and most UNIX systems on the same hardware (even multiprocessor boxes). Linux has been chosen by hundreds of thousands of ISPs⁵, by many university computer labs, and surprisingly by many businesses—by all people that need reliable server and network performance in many different situations.

Linux supports all of the most common Internet protocols, including FTP⁶, POP, SNMP⁷, and many more. Linux can operate as a client or as a server for all of the above and has already been widely used and tested in the above roles.

local area network
局域网
swap /'swɒp/
n. & v. 交换
swap space
交换空间

Linux also fits easily and tightly into your **local area network**, no matter what combination of systems you might be running. Full and seamless support for Macintosh, DOS, Windows, Windows NT, Windows 95, Novell, OS/2, all using their own native protocols—Linux can do all of this in just 16MB of memory or even less (with **swap space**). Typically, you can have all the above running in 8MB with 16MB swap (a total of 24MB RAM).

blinding /'blaɪn dɪŋ/
a. 眩目的, 使人眼花缭乱的
source code
源(代)码
kernel /'kɜ:nəl/
n. (果核或果壳内的)仁; 核心

Linux is developing at a **blindingly** fast speed, and ports exist to PowerPC⁸, Macintosh, and so on. If Linus Torvalds ever does decide to abandon the project, since we have the full **source code** available, he can just assign somebody else to take his place. The Linux **kernel** development will still continue no matter what happens to Linus. Somebody else will just continue with it.

Exercises

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

-
- ¹ X.25: 一种专用网络, 使用包交换协议, 绕过有噪音的电话线路传输数据。
² ISDN: 综合业务数字网, 俗称一线通 (Integrated Services Digital Network 的首字母缩略)。
³ PLIP: 并行线路接口协议 (Parallel Line Interface Protocol 的首字母缩略)。
⁴ Novell: Novell 网络操作系统平台。美国著名的网络产品开发商 Novell 公司的产品。
⁵ ISP: 因特网服务提供者 (Internet service provider 的首字母缩略)。
⁶ FTP: 文件传输协议 (File Transfer Protocol 的首字母缩略)。
⁷ SNMP: 简单网络管理协议 (Simple Network Management Protocol 的首字母缩略)。
⁸ PowerPC: PowerPC 微处理器, 一种由 Motorola 公司和 IBM 公司于 1992 年开发的微处理器体系结构, Apple 公司也参与了部分开发工作。

1. Linux is a free _____ clone developed by Linus Torvalds and a team of programmers over the Internet.
2. Since it was developed through cooperative efforts over the Internet, _____ support in Linux is superior to most other operating systems.
3. The features contained in Linux include true _____, virtual memory, the world's fastest TCP/IP _____, shared libraries, and multi-user capabilities.
4. According to the text, Linux can operate as a(n) _____ or as a server for all of the most common Internet _____, including FTP, POP, SNMP, and so on.

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

- | | |
|------------------|-------------|
| 1. serial port | 5. 虚拟内存 |
| 2. ISDN | 6. 源代码 |
| 3. token ring | 7. 交换空间 |
| 4. fast Ethernet | 8. 超级视频图形阵列 |

Unit 4: Computer Language and Programming

(计算机语言与编程)

Section A

Programming Language

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 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

syntax /'sɪntæks /

n. 句法

ambiguous

/æm'bigjuəs /

a. 含糊不清的,
模棱两可的

punctuation

/,pʌŋktju'eɪʃən /

n. 标点符号

sophistication

/sə'fɪstɪ'keɪʃən /

n. 复杂性; 尖端性

versatility

/,vɜ:sə'tɪlɪti /

n. 多用途; 通用性

address /ə'dres /

v. 对付, 处理

portable /'pɔ:təbəl /

a. 可移植的;
便携的

¹ **FORTRAN**: FORTRAN 语言, 公式翻译程序语言 (Formula Translation 的缩合)。

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

solve diverse types of computing problems. Languages like C, PASCAL¹, and BASIC 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. 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++, PASCAL, and FORTRAN. **Assembly languages** are **intermediate languages** that are very close to machine languages and do not have the level of **linguistic** sophistication exhibited by other high-level languages, but must still be translated into machine language.

assembly language

汇编语言

intermediate language

中间语言, 中级语言

linguistic

/liŋ'gwɪstɪk/

a. 语言(学)的

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.

storage register

存储寄存器

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

statement

/'steɪtmənt/

n. 语句

¹ PASCAL: PASCAL 语言。

compiler

/kəm'paɪlə/

n. 编译程序,
编译器

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 the 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 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

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

¹ *in essence*: 本质上, 实质上。

functional language

函数式语言

rigorous /'rɪgərəs/

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.

outgrowth

/'aʊtgrəʊθ/

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 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.

hierarchy

/'haɪərə:ki/

n. 层次, 分层 (结构), 分级 (结构)

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:

deduce /di'dʒu:s/

v. 推论, 推断

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

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

macro /'mækrou/
n. 宏, 宏指令

data declaration
数据声明

array /ə'rei/
n. 数组, 阵列

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
/pə'ræmitə/
n. 参数, 参量

functions work. Programmers use statements to manipulate common components of programming languages, such as variables and **macros** (mini-programs within a program).

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.

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. 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 Backus¹, and then COBOL, created by Grace Hopper². The first functional language was LISP³, written by John McCarthy⁴ 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 SIMULA⁵, emerged. Logic languages became well known in the mid 1970s with the introduction of PROLOG⁶, a language used to program artificial intelligence software. During the 1970s, procedural languages continued to develop with ALGOL⁷, BASIC, PASCAL, C, and Ada⁸. SMALLTALK⁹ was a highly influential object-oriented language that led to the **merging** of object-oriented and procedural languages in C++ and more recently in JAVA¹⁰. Although pure logic languages have declined in popularity, variations have become vitally important in the form of **relational languages** for modern databases, such as SQL.

merge / mɜːdʒ /
v. 合并; 结合

relational language
关系语言,
相关语言

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 machine language is a(n) _____ language in binary code that the 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.

¹ *John Backus*: 约翰·巴克斯 (1924~), 美国计算机科学家。

² *Grace Hopper*: 格雷斯·霍珀 (1906~1992), 美国应用数学家, 计算机程序语言的开拓者。

³ *LISP*: LISP 语言, 表处理语言 (*List Processing* 的缩合)。

⁴ *John McCarthy*: 约翰·麦卡锡 (1927~), 美国计算机科学家。

⁵ *SIMULA*: SIMULA 语言, 仿真语言, 模拟语言 (*Simulation Language* 的缩合)。

⁶ *PROLOG*: PROLOG 语言, 逻辑程序设计语言 (*Programming in Logic* 的缩合)。

⁷ *ALGOL*: ALGOL 语言, 算法语言 (*Algorithmic Language* 的缩合)。

⁸ *Ada*: Ada 语言。

⁹ *SMALLTALK*: SMALLTALK 语言, IBM 开发的面向对象的编程环境 (语言)。

¹⁰ *JAVA*: Java 语言, 一种新型的计算机语言, 可以跨平台运行。

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.

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. SQL | 20. 计算机程序设计员 |

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

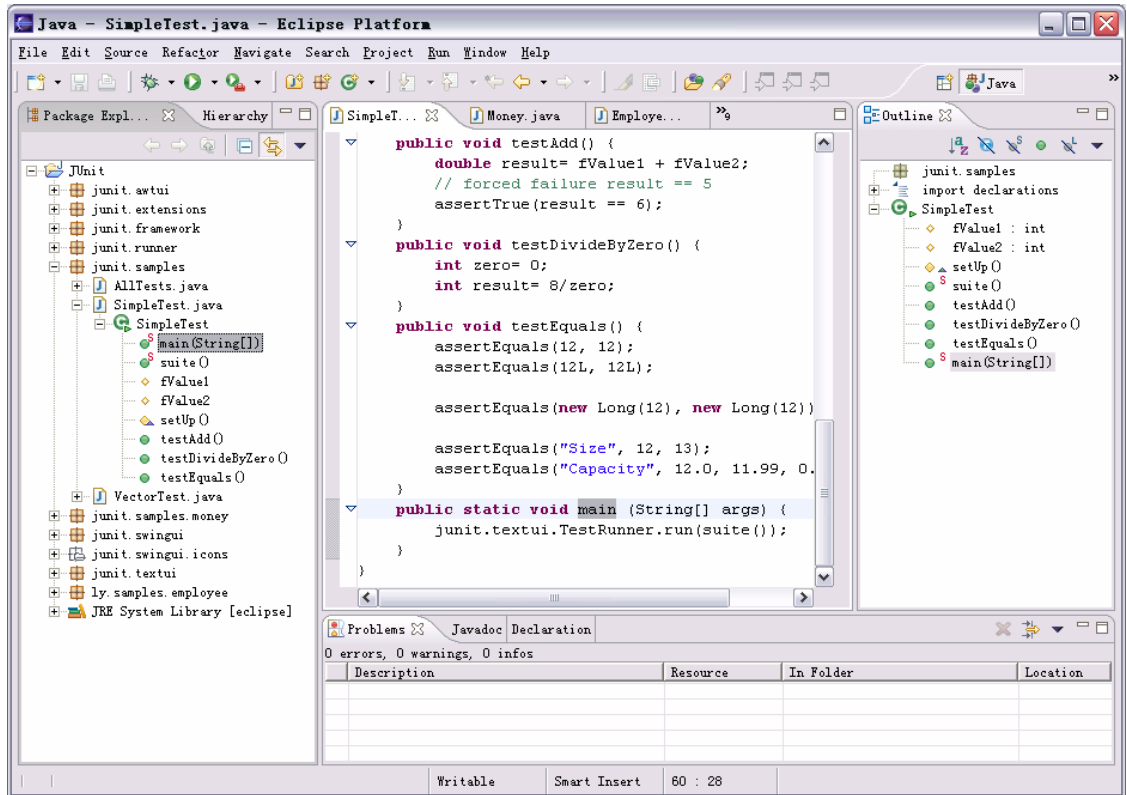
_____ to use *mnemonics* (助记符) for instructions and symbols for variables. Such programs are then _____ by a program known as an assembler into the binary encoding used by the _____. Other pieces of system software known as *linking loaders* (连接装入程序) _____ pieces of assembled code and load them into 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:

One especially powerful feature of OOP (object-oriented programming) languages is a property which is known as inheritance. Inheritance allows an object to take on the characteristics and functions of other objects to which it is functionally connected. Programmers connect objects by grouping them together in different classes and by grouping the classes into hierarchies. These classes and hierarchies allow programmers to define the characteristics and functions of objects without needing to repeat source code, the coded instructions in a program. Thus, using OOP languages can greatly reduce the time it takes for a programmer to write an application, and also reduce the size of the program. OOP languages are flexible and adaptable, so programs or parts of programs can be used for more than one task. Programs written with OOP languages are generally shorter in length and contain fewer *bugs* (缺陷), or mistakes, than those written with non-OOP languages.

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:'pæ:fluəs/
a. 多余的; 过剩的

instantaneous

/.instən'teinjəs/
n. 瞬间的, 即刻的
overhead

/'əuvəhed/
n. 开销; 经常 (或管理、间接) 费用

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**.

clutter /'klʌtə/
n. 凌乱, 杂乱; 杂乱的东

cryptic /'kriptik/
a. 隐含的; 意义含糊的

header /'hedə/
n. 页眉; 标头

preprocessor
/pri:'prəʊsesə/
n. 预处理程序, 预处理器

arcane /ɑ:'keɪn/
a. 神秘的, 晦涩难解的; 秘密的

delve /delv/
v. 搜索, 翻查

reference
/'refərəns/
v. 引用; 参照

software package
软件包

directory
/di'rektəri/
n. 目录

revolve /ri'vɒlv/
v. 旋转; 围绕

class hierarchy
类层次

encapsulation
/in.kæpsə'leɪʃən/
n. 封装

polymorphism
/ˌpɒli'mɔːfɪzəm/
n. 多态性, 多形性

clarity /'klærəti/
n. 清晰, 明晰

You will have the feeling that you have eliminated a lot of **clutter** from 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** API³ 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. I know I did.

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

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。

it. To write any sort of Java program, no matter how simple, you must build 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.

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 around.

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.

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.

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 right"? The Java **linker** understands class types and performs

tailor / 'teɪlə /
v. 针对特定目的
(或对象)作修改,
使适应(特定需要)
temptation
/ temp'teɪʃən /
n. 引诱(物), 诱惑
(物)

explicit / ik'splɪsɪt /
a. 明确的; 详述的
assembly code
汇编码

dinosaur
/'daɪnəsɔː /
n. 恐龙; (尤指废弃
过时的) 庞然大物

runtime / 'rʌntaɪm /
n. 运行时刻
syntactic
/ sɪn'tæktɪk /
a. (按照) 句法的
semantically
/ si'mæntɪkəli /
ad. 在语义上

mom / mɒm /
n. <主美口> 妈妈
linker / 'lɪŋkə /
n. 连接程序, 链接
程序

bound /baund/
n. [常作~s] 边界; 界限, 界线

bounds checking
边界检查

hacker /'hækə/
n. 黑客

flaw /flɔ:/
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

offense /ə'fens/
n. 冒犯; 攻击

applet /'æplət/
n. 小应用程序

Trojan (horse) /'trɔ:dʒən/
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

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

/'mʌlti,θredɪd /

a. 多线程的

multithreading

/'mʌlti,θredɪŋ /

n. 多线程操作

thread /θred /

n. 线程, 线索

duke /dʒu:k /

n. 公爵

seemingly /'si:miŋli /

ad. 表面上; 看来

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'twain /

v. (使) 缠结, (使) 缠绕在一起

concurrently

/kən'kʌrəntli /

ad. 同时发生地, 并发地, 并行地

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; 产生

native code

本机代码, 原生代码

underway

/'ʌndə'wei /

a. 在进行的中的

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

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/
a. <拉> 实际的, 事实上的
offshoot /'ɒfʃu:t/
n. 支族, 旁系; 衍生事物

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.

broker /'brəʊkə/
n. 代理者; 代理程序

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 4C-1.

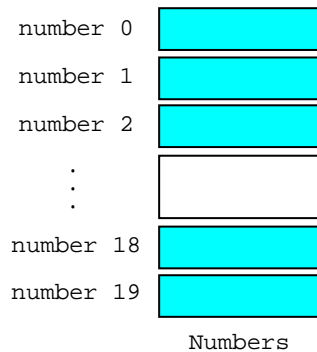


Figure 4C-1: Twenty individual variables

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 4C-2 to read, process, and print these 20 numbers.

flowchart
/ˈfləʊtʃɑ:t/
n. 流程图

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

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

/'deizneɪt/

v. 标出,把...定名为

subscript

/'sʌbskrɪpt/

n. 下标,角注

ordinal /'ɔ:dɪnəl/

a. 顺序的

ordinal number

序数

designate the first element number_0 as shown in Figure 4C-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

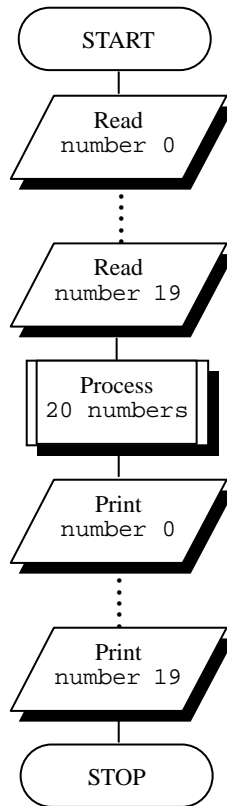


Figure 4C-2: Processing individual variables

What you have seen is that the elements of the array are individually addressed through their subscripts (Figure 4C-3).² The array as a whole has a name, `number`, but each member can be accessed individually using its subscript.

programming**construct**

编程结构

The advantages of the array would be limited if you didn't also have **programming constructs** that allow you to process the data more

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。

loop /lu:p/
n. & v. 循环 (体)

conveniently. Fortunately, there is a powerful set of programming constructs, **loops**, that makes array processing easy.

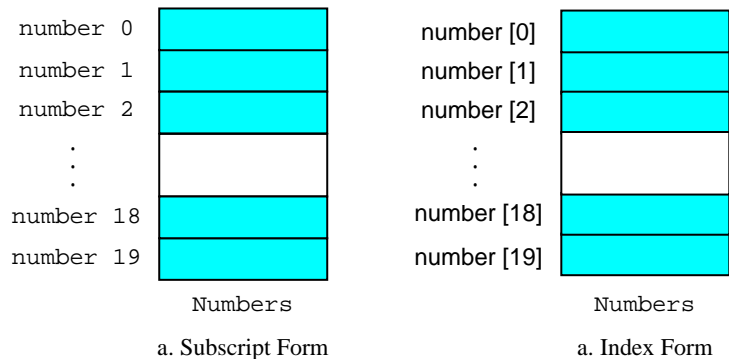


Figure 4C-3: Arrays with subscripts and indexes

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 number_0 as $\text{number}[0]$.

Following the convention, number_1 becomes $\text{number}[1]$ and number_{19} becomes $\text{number}[19]$. 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 4C-4.

In this 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 /'ɪndeksiŋ/
n. 编索引; 标引;
变址; 加下标
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。

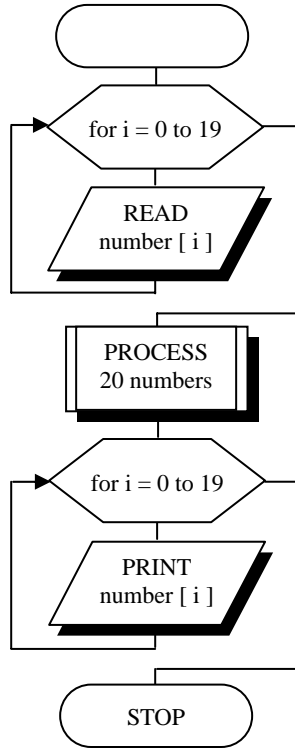


Figure 4C-4: Processing an array

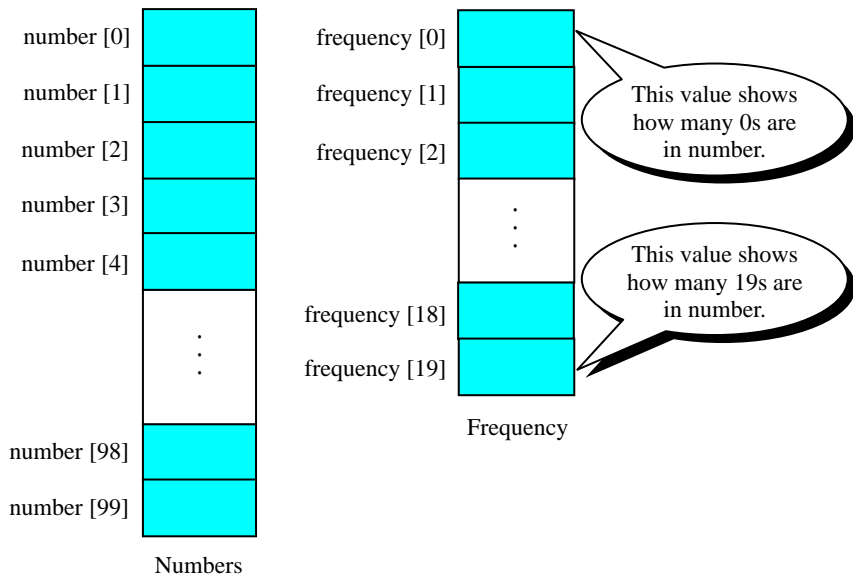


Figure 4C-5: Frequency array

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.

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 4C-5).

histogram

/ˈhɪstəgræm/

n. 直方图, 矩形图;
频率分布图

pictorial /pɪkˈtɔːriəl/

a. 图示的; 图像的

bar /bɑː/

n. 条形图

bar chart

条形图

asterisk /ˈæstərɪsk/

n. 星号

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 4C-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.

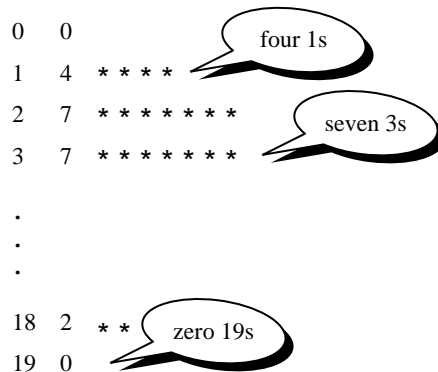


Figure 4C-6: Histogram

one-dimensional

array 一维数组

linear /ˈlɪniə/

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 4C-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

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。

memory.¹ If you were to consider memory as a row of bytes with the 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 4C-8).

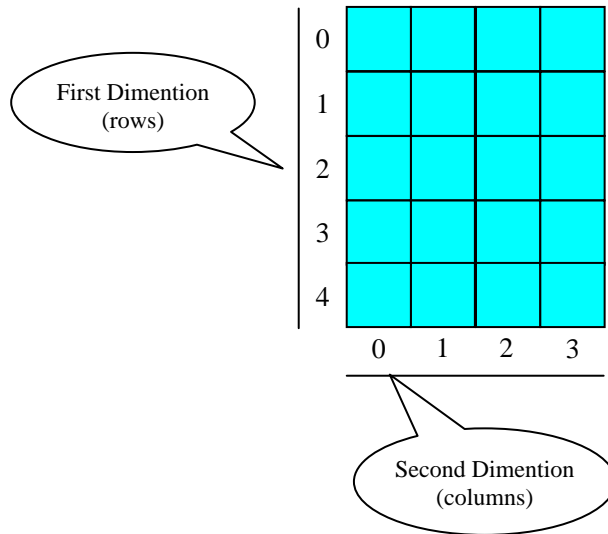
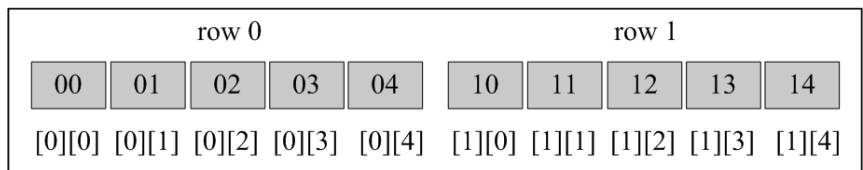


Figure 4C-7: Two-dimensional array

view / vju: /
n. 视图

00	01	02	03	04
10	11	12	13	14

User's View



Memory View

Figure 4C-8: Memeory layout

¹ 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。

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 |