

软件半月通

HOW SOFTWARE WORKS

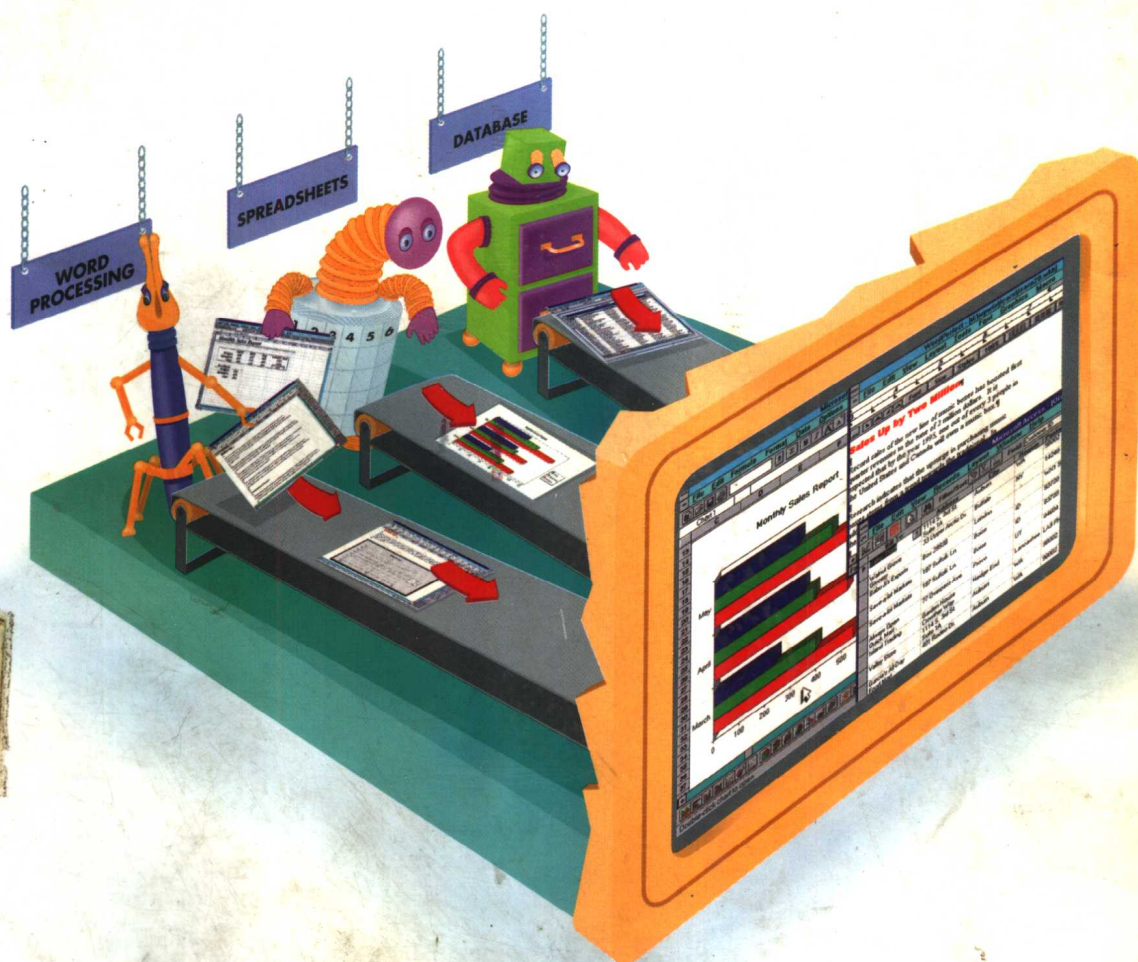
RON WHITE

Author of the National Bestseller *PC/Computing HOW Computers WORK*

Illustrated by PAMELA DRURY WATTENMAKER

郭飞 译

郭成忠 审校



電子工業出版社

How Software Works 软件半月通

软件半月通

How Software Works

原作者 Ron White

郭 飞 译

郭诚忠 审校



電子工業出版社

(京) 新登字055号

内容提要

本书以通俗易懂的语言、赏心悦目的精美图示,把软件如何在计算机中工作展现在读者面前。

作者深入浅出地介绍了软件与硬件如何协调工作;程序语言如何把人的思想翻译成软件;字处理软件如何形成文件;通信软件如何控制调制解调器实现通信……等等。本书覆盖了所有种类的软件,包括操作系统、数据库管理软件、页表处理、字处理、通信和窗口软件等。全书共分二十二章,配有精美插图,是广大计算机软件初学者的良师益友。



Copyright © 1993 by Ziff-Davis Press. All rights reserved.

Ziff-Davis Press and ZD Press are trademarks of Ziff Communications Company.

本书英文版由美国Ziff-Davis Press出版, Ziff-Davis Press已将中文版独家版权授予北京美迪亚电子信息有限公司。未经许可,不得以任何形式和手段复制或抄袭本书内容。

软件半月通

How Software Works

Ron White

郭飞译

郭诚忠审校

责任编辑 吴晓燕

*

电子工业出版社

电子工业出版社发行 各地新华书店经销

北京市顺义县天竺颖华印刷厂印刷

北京美迪亚电子信息有限公司排版

开本: 787×1092 毫米 1/16 印张: 10 字数: 240 千字

1994年3月1版 1994年3月第1次印刷

印数: 10100册 定价: 22.00元

ISBN 7-5053-2340-7/TP·660

出版说明

计算机科学技术日新月异，为了引进国外最新计算机技术，提高我国计算机应用与开发的水平，中国电子工业出版社与美国万国图文有限公司合资兴办的北京美迪亚电子信息有限公司取得了美国 Ziff-Davis Press 的独家版权代理。Ziff-Davis Press 授权本公司通过电子工业出版社等出版机构全权负责在中国大陆出版该公司的中文版和英文版图书。现在与广大读者见面的是最近推出的第一批图书。今后我们还将陆续推出 Ziff-Davis Press 的最新计算机图书和软件，为广大读者提供更好的服务，传递更多的信息。

美国 Ziff-Davis Press 是全美最大的计算机出版商之一，它出版的书籍、杂志和光盘，主办的展览和会议，提供的咨询和网络服务，形成了整个行业潮流的主导。我们优选翻译出版的图书是 Ziff-Davis Press 的最新计算机图书，并采用了该公司提供的电子排版文件，从而提高了质量并大大缩短了图书的出版时间，从根本上改变了以往翻译版图书要落后原版书较长的“时差”现象，这在电子技术日新月异的时代具有深远的意义。

北京美迪亚电子信息有限公司

1994年3月

简介

大多数人以前从未看见过计算机软件的源代码，这些代码甚至比神仙的咒语还难理解。当然，魔术似乎太简单，简单的东西就不能强有力，可是软件却是人类创造的最强大的工具之一。但是，对很多人来说，它仍然是个魔术。当把一张软盘放入个人计算机，或为计算机打入一个程序名，或者用鼠标指点一个文件，突然所有这些东西开始发生，从个人机上发散出美丽的彩色图象，声音和语言，它们活现在我们面前，也许只能用魔术来形容。软件看起来只是一系列的数字，并且很难预测它今后三个月之内的发展。当你询问有关人、国家、数据信息时，软件的回答更像一个水晶球。如果你问软件把你带到很远的电子广告牌那里，你就像坐在魔毯上一样，只需一秒钟就到了那里。

软件看起来既强大又像全套咒文，它可能使我们每个人变成魔术师的徒弟。这本书不想把你变成一个程序员，但是，即使不学习程序代码这个怪物，你也能学到这个神秘工作的基础知识和为什么它那样强大，并产生对这个神秘工作和创造人员的尊重。同样，你将能较好的使用这些神厅的概念，进入你的软件，字处理和图象处理，程序代码会在你的屏幕上引导出真实的结果。

谈到计算机和软件总是有一点让人担心，这种担心总是伴随着我们不懂的东西。这也是本书为什么想帮助你理解软件如何工作的原因，而且，不可避免地，就像在此描述的那样，是比较复杂的。恕我冒昧，采用过于简单的方法叙述，以便集中于大事情，防止在细节上混淆不清。那些对软件有较深了解的人，可以给出解释，即本书只是含概了某些软件如何工作的一个领域。我还要说什么呢！我不能争论说软件不是复杂的，不是微妙的。但是，我要强调，这本书要告诉你，要了解这个复杂的神密变化过程，什么才是最重要的。

然而，大量地简化，只是限于DOS和Windows基础上的软件。把所有MAC的热心者均排除在外，我只能说，在原则上，他们软件如何工作，基本上同DOS PC的软件工作是相同的，但是，为了避免把DOS和MAC系统无休止的比较，我选择了只集中于操作环境的叙方法，对此，我也比较熟悉。

综上所述，我希望读者会发现这些信息是娱乐性的，也很有用。软件学习是一次智能的锻炼，比任何我所遇到的事情更鼓舞人。我希望你能分享我对程序和程序员所有这些意图的乐趣。

目 录

引言



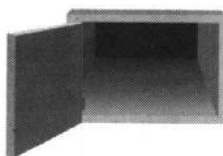
1 第一部分 硬件和软件如何协同工作

- 4 第一章
微处理器怎样运行软件
- 10 第二章
BIOS 如何与软件协同工作
- 14 第三章
操作系统与BIOS 怎样交互式工作
- 26 第四章
存储器管理程序如何工作

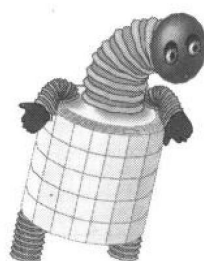


35 第二部 程序语言如何工作

- 38 第五章
机器语言如何工作
- 43 第六章
编程语言如何把思路转化为软件
- 48 第七章
语言解释器和编译器如何工作

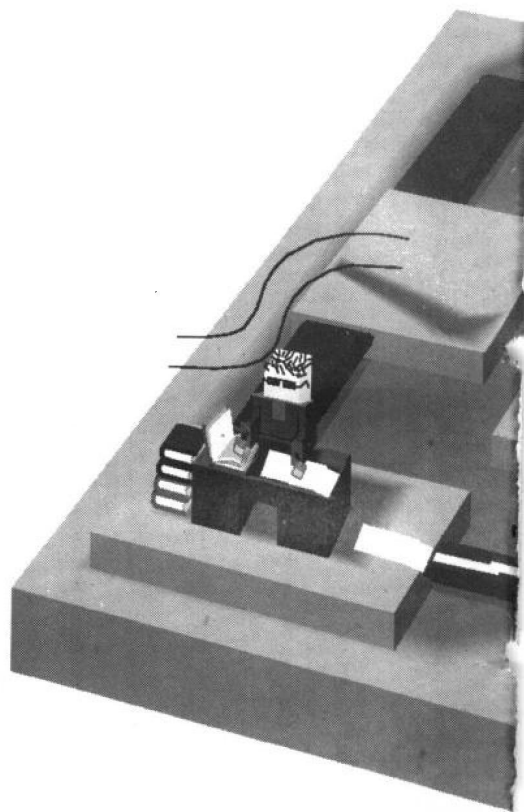


- 55 第三部分
数据库如何工作
- 58第八章
数据库如何存储信息
- 62 第九章
数据库索引如何工作
- 66第十章
关系型数据库如何工作



71 第四部分 电子表软件如何工作

- 73 第十一章
电子表如何存储数据
- 77 第十二章
电子表的计算公式如何工作

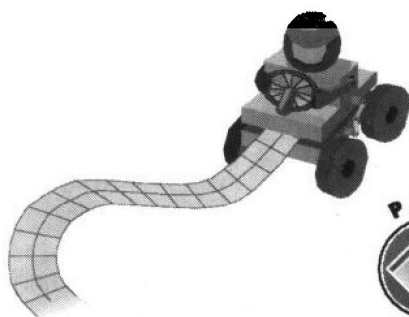




81 第五部分
文字处理程序如何工作

84 第十三章
文字处理的文件是怎样编
排的

90 第十四章
字型如何工作



111 第七部分
通信软件如何工作

112 第十七章
通信软件如何操作调制解
调器

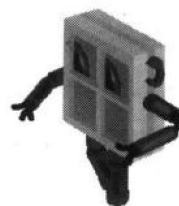
118 第十八章
数据传输协议如何工作



97 第六部分
图形软件如何工作

100 第十五章
位图图形学如何工作

107 第十六章
矢量图形如何工作

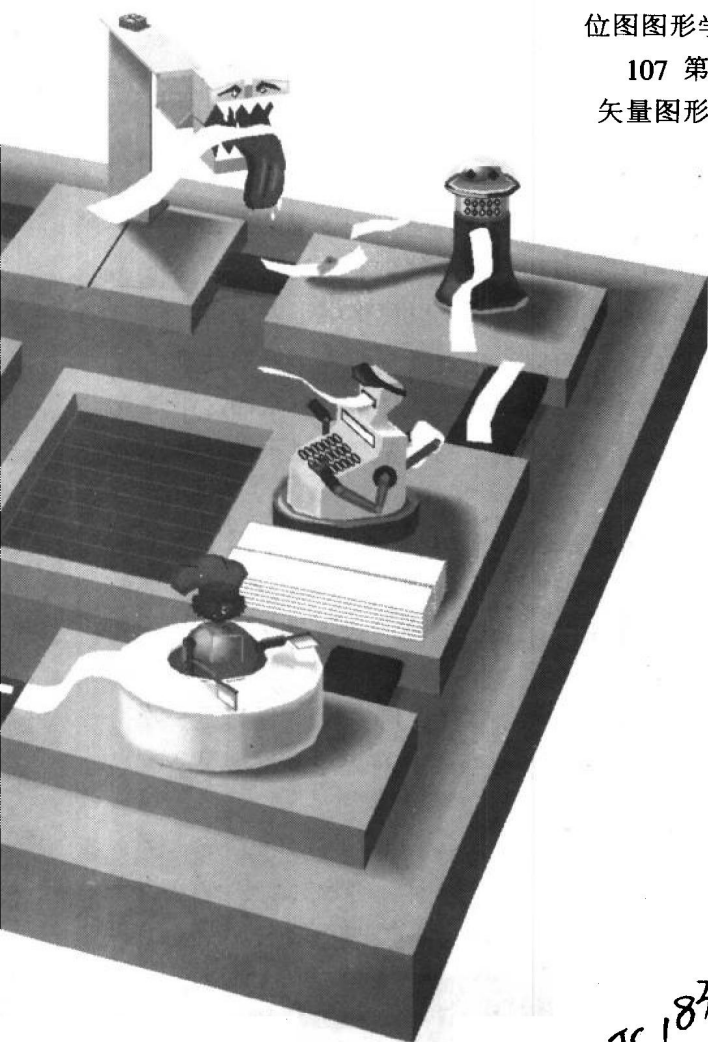


123 第八部分
Windows如何工作

126 第十九章
Windows的图形界面如何
工作

133 第二十章
Windows如何运行多进程
138 第二十一章
Windows如何共享程序代
码

143 第二十二章
Windows如何共享数据



075183/09



第一部分 硬件和软件如何协同工作

即使你以前从未接触过计算机，你也一定用过软件。就如WordPerfect和Lotus1-2-3就是软件，音乐唱片和录象带也是软件。一个程序是控制一系列有序操作的一组指令。

一张哈蒂斯大婶的胡萝卜饼菜谱、一件衣服的样式和一个电话号码等都叫程序。它的表现形式就如同计算机代码的打印清单或烹调书的一张菜谱，即那些不能被硬件直接读取的东西。一个人或一台机器每当使用程序时，就是在执行在程序中描述的一系列动作。软件是程序的一种特殊形式，它以一定的形式记录下来，或记录在计算机软盘上或记录在音像带上，这样，程序每次执行时就不必用手工方法输入到硬件中去。

任何形式的程序和软件都有一个共同点，就是指示硬件如何做有用的事情。一部电唱机（或更现代一些的CD唱机），一台录象机，一个火炉和炊具，一台缝纫机和一部电话机都叫硬件，硬件是有形的实体，实在的工具，只能通过编程或软件才能赋予它生命，否则它不能自己动作，也就没有多大用处。

举个例子，一个羊角锤本身与镇纸相比恐怕没有更多的用处。但是，当一个木匠使用它时，它可以钉钉子，拔出旧钉子，甚至砸开胡桃。木匠挥动锤子就是对运动中的锤子进行编程。

有些硬件，如你收集的炊具、火炉、缝纫机、电话机，在操作时都需要比挥动锤子更复杂的人的参与。烹调、缝纫，给哈蒂斯打电话问她为什么你的胡萝卜饼不象她做的那样，这些过程与早期为计算机进行精确有序的手工编程是很类似的。早期的计算机大部分被那些有强烈拥有欲的爱好者所购买。那些PC机既没有键盘也没有显示器，它们的拥有者饶有兴趣地通过精确、顺序地拨动开关的过程来编程，然后根据机器前端的指示灯图案来读出计算的结果。那些计算机爱好者所做的与你在微波炉时所做的是类似的，你以某个次序按下按键，使得微波炉能够按照一个特定的功率水平和时间限制来工作，这就是编程。

如果你按下预设置微波炉工作状态的按键，那么说明你正在使用软件，这也是一个可以预配置的指令集。从这方面来说，这些指令被永久地记录在微波炉内部的微型芯片当中。从一个录音设备和录象机获取的信号，是一种软件，它告诉激光唱机和录象机应当把什么样的电脉冲发送到扬声器或电视机以再现Mahler的第十交响曲的音乐或《终

结者Ⅱ》的视频效果。

在个人计算机系统中，任何你能看见的、摸得到的、感觉到的，或能体味到的（如果你喜欢这样做话）都是硬件，如芯片、显示器、键盘、打印机、调制解调器。软件是无形的，你可以抓住记录计算机软件的媒体，如一张软磁盘或者一块硬盘，但是软件本身是一个抽象的指令集，它可以引导硬件完成特定的、具体的任务。

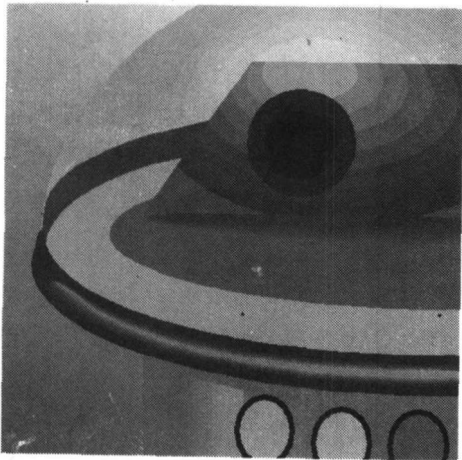
严格来讲，程序工作时必须针对特定的硬件。如果你把录象机录制的信号传送给录音磁带，录音设备在读出那些信号的时候只能产生毫无意义的噪声。与此类似，一个为IBM兼容的PC机编写的程序如果搬到MAC机上，也同样毫无意义，什么事也不会发生。

在硬件和多数软件之间是操作系统，操作系统实际上是另一种软件程序，但属于一个特殊类型。它不允许你写信、计算预算或者跟踪一个存货清单报表，但它允许你编程，与硬件程序一起工作，完成预定任务。

没有操作系统是可以编写软件的，这类软件把指令直接发送到微处理器和PC的其它硬件单元。但操作系统完成两个任务，使编程者做起来更容易和让你的程序在相同类型的PC上都能运行。这些任务是非常重要的。首先，程序员编写的应用软件无需与硬件直接对话，操作系统减轻程序员大量相同的乏味工作，如果没有操作系统，每个编程者将不得不自己解决程序在屏幕上显示文本或图形的问题、程序同打印机发送数据的问题、程序在硬盘上读写数据的问题、以及需要软件、硬件紧密结合的其它多种功能的问题。

一个操作系统所做的就是要让程序员操作起来更容易。它确保同一程序在不同的PC上以同样的方式运行。它遮掩了不同的两种机器硬件间的差异，所以同一应用程序可在多种机器上运行。

在<<How Software Works>>这部书中，我们实际上在应用软件上只花了很少一点时间，我们将着重讲解微处理器和操作系统及它们最基本的工作方式，其应用程序如何工作。这些要素不仅影响到你的PC系统的结构，还关系到软件应用操作时的能力和速度。





第一章 微处理器怎样运行软件

个人计算机的微处理器由成万甚至上百万个叫做晶体管的微小通/断开关复合而组成。晶体管沿着芯片上的电路安装，这些电路是由超细的铝线构成的。当晶体管以一定的模式安排好之后，微处理器的某一部分就被指定用来存储数据，其它部分则能以不同的方式操作那些数据。

晶体管的开/关状态使得它们便于表示二进制数，使数据和软件代码得以存储。在二进制系统中，只有两个数字，0和1，他们可以表示所有的数。十进制的1在二进制中也是1，但是2在二进制中是10，3是11，4是100，5是101，6是110，7是111等等。一个晶体管被断开表示0；晶体管被接通表示1。每个信号0或1被称做位（Bit），八个位组成一个字节（byte），1024个字节组成1K（Kilobyte）。

计算机把软件代码和数据都看成是一系列的二进制数。如果你能把自己缩小到一个细胞的大小，进入到芯片内部，就可以看到微芯片中的晶体管是怎样组成个人计算机的随机访问存储器（RAM）的，它怎样被接通或断开，你会发现大量变化不停的0和1，但你无法确定阵列中的哪一部分用来表示代码，哪一部分表示数据。

Intel的80386，80486，以及Pentium 微处理器和他们的仿造者，都是32位处理器，这意味着他们能处理的二进制数可高达32位宽度。最大的32位二进制数是1111 1111 1111 1111 1111 1111 1111 1111，转换成十进制数就是4, 294, 967, 295。执行计算任何比这个数大的数，需要把它分成小的部分，完成对每个部分的数学操作，然后把结果组合起来。4, 294, 967, 295这个数字还有另一种意义，即所有的存储器都以地址来识别，4, 294, 967, 295表示处理器能够寻址的所有存储器的最大限度。在PC内存芯片中数据的每一字节都被编号，以建立一个物理地址或实态地址，表示在RAM或ROM芯片中由实际晶体管组成的那个字节的位置。由于应用软件不能知道在每个PC内存中的物理地址，软件使用了一个不同的编址方案，称做逻辑地址。微处理器结合PC的操作系统，运用逻辑地址，使物理地址更加符号化，在读写数据的时候把逻辑地址翻译成物理地址。

尽管现代处理器能寻址4兆字节以上的内存，大多数的PC机当必须使用640K以上内存时，还须依赖操作系统管理。最初的IBM PC机使用的处理器为Intel 8088，是16位

的处理器，也就是说，它能操作长度最大为16位的数据，那意味着内存空间地址仅能是1111 1111 1111 1111（16个位的值均置为1），这使得8088仅能直接使用64K内存。

若想使更多的RAM成为可用的，为8088而设计的操作系统，MS-DOS的开发者设计了一个分段的内存方案，它组合了两个16位2进制数，以这种方式产生一个20位地址，用这种方法，DOS使得8088可以寻到1,024K字节，或1megabyte节（MB）内存（RAM），然而对于1MB内存，只有640K被DOS运行程序使用，剩下的内存留给多种需要内存地址的硬件设备。

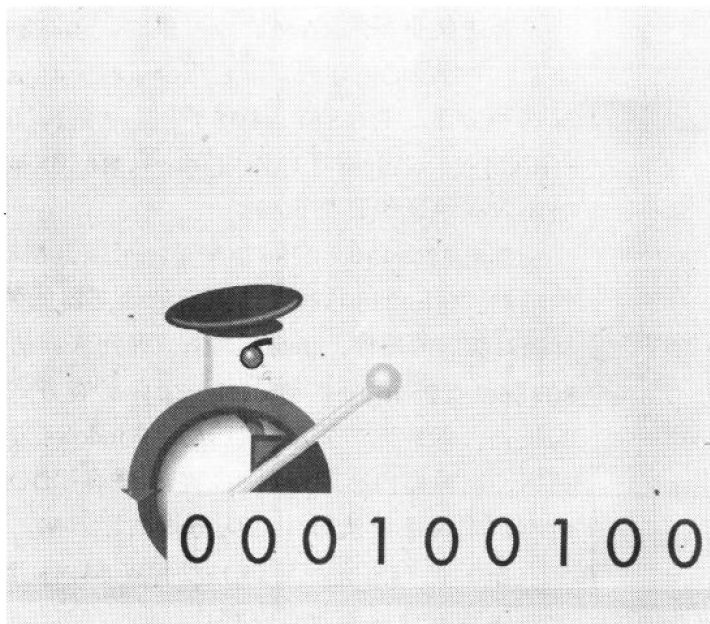
继8088之后的80286处理器，可以直接寻址到16MB的内存，但仍受DOS内存寻址的限制。Intel的80386以及更新的处理器能使用更多的内存，但当使用实态方式时，同样面对相同的限制，所谓实态方式是用来模仿8088芯片工作的一种操作方式，（80386，80486以及Pentium芯片还有一个保护态方式，它充分使用其内存直接寻址和特殊指令的优点。保护态方式需要OS/2，Windows NT或一些其它高级操作系统），在实态方式下，被称作内存管理器的特殊程序有如DOS的一个延伸，它允许处理器使用以兆计的内存，同时继续使用DOS的分段内存方案。

新的操作系统，诸如OS/2和Windows NT，使用平面内存方案，能够直接寻址32位处理器能寻址的所有内存。

在这一章中，我们将看到分段内存方案是如何工作的，以及80386微处理器如何使用存储在内存中的软件代码和数据。Intel 80386微处理器在这里用作例子，因为它的寻址方式和内存指令系统的发展是一个有意义的转折。早期的处理器不包含80386中特殊功能部件，更新的80486及Pentium微处理器与80386在概念上相同，但它包含一些特殊部件，设计用来进行浮点数值计算和加速代码与数据在芯片中的传输速度。

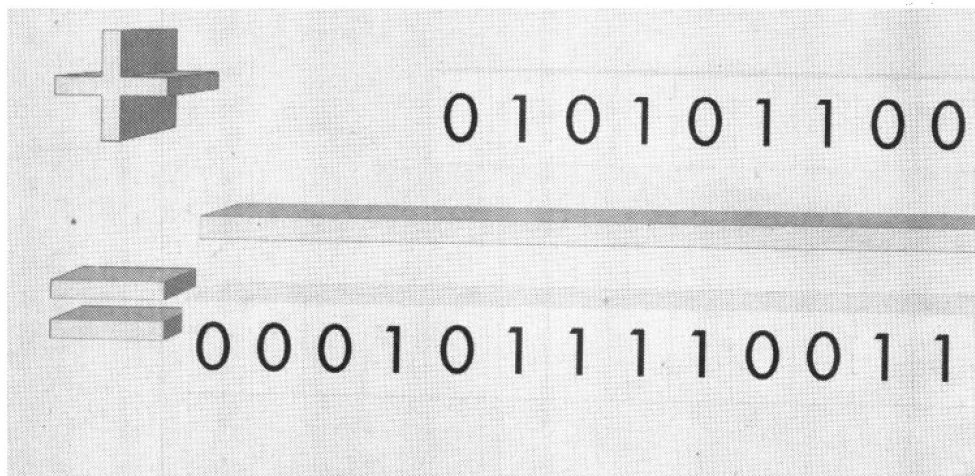
DOS怎样使用分段内存

- ①若想访问某一存储器单元，DOS首先把一个16位物理地址装入微处理器的某个寄存器当中，该寄存器是微处理器中一个特定的晶体管存储单元，用于在处理器操作数据的时候保持数据。



- ②处理器将16位地址左移4位，这就好像乘以一个16进制数一样，或在它后面加4个0形成一个20位二进制数。有了额外的4个0，这个20位数可以表示每个16位存储器单元中的一个，即其20位地址末尾是4个0的那个单元。这就是段地址。

00010010000110



- ③处理器在段地址上再加上一个16位数字，也就是相关地址，或偏移量。其结果就是有了一个完整的20位数，表示一个线性地址。

注：OS/2和Windows NT 使用平板内存方案，可以直接寻址到4G RAM，而无需像DOS那样处理地址。

④处理器定位一个内存地址时，首先查找段地址，然后查找段地址的偏移量。

1 1 0 1 0 0

0 0 0 0 0 0

1 1 1 0 0 0

1 1 1 0 0 0

	0001h	0002h	0003h	0004h	5676h	5677h	5678h	5679h	5680h	5681h	5682h	5683h
1232h												
1233h												
1234h												
1235h												
1236h												
1237h												

⑤然后处理器根据段地址和偏移量控制电路找到与特定的逻辑地址相对应的RAM晶体管芯片的实际（物理）单元。

80386微处理器如何运行软件

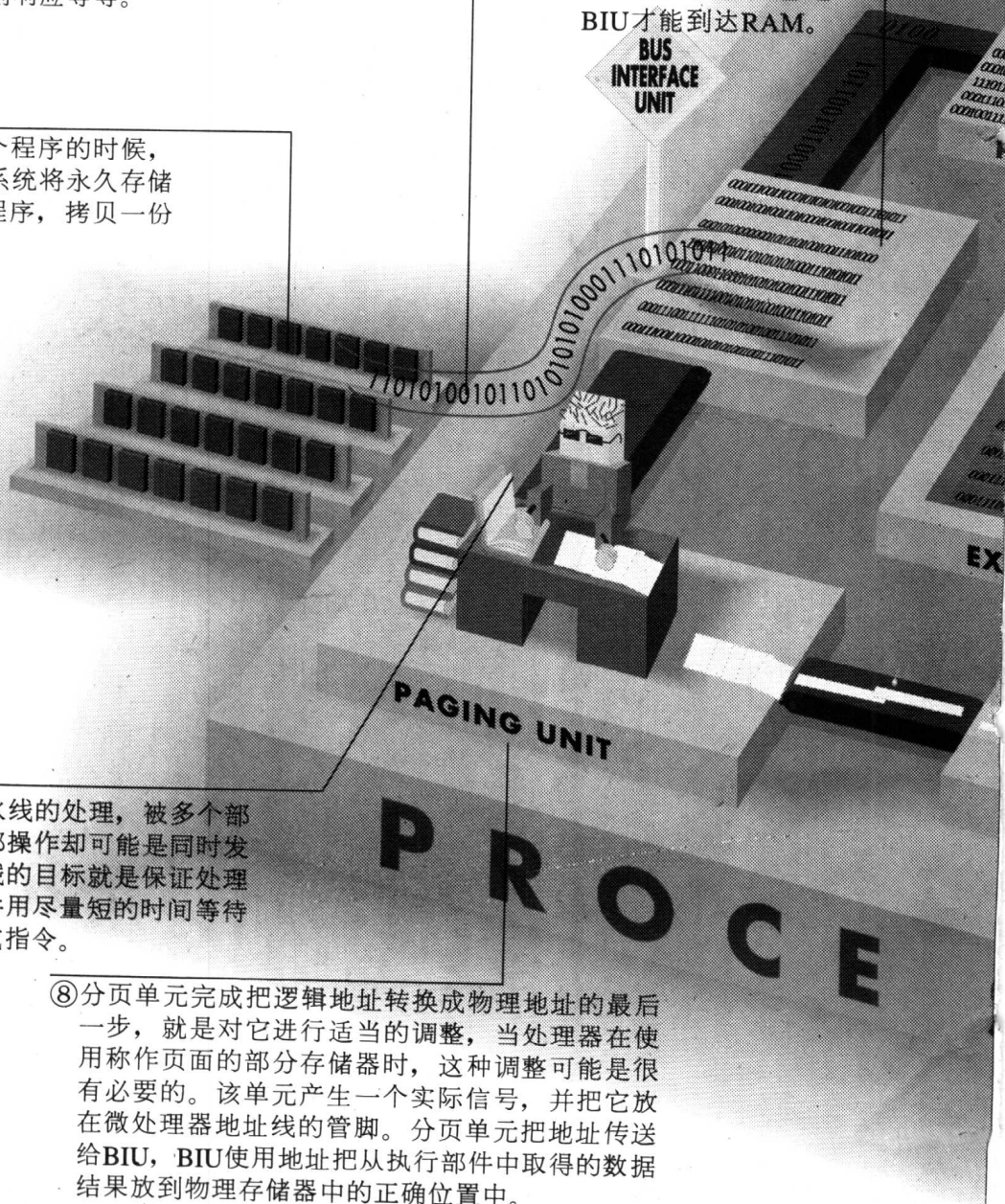
②微处理器从RAM中程序的第一个字节开始读程序的代码。它读每个连续的字节并执行包含在代码中的指令。这样持续到有以下两种事件之一发生：微处理器遇到一条指令让它跳过、循环（loop）、或者转到程序的另一单元；由软件或硬件设备产生一个中断信号，诸如键盘要求处理器立刻响应等等。

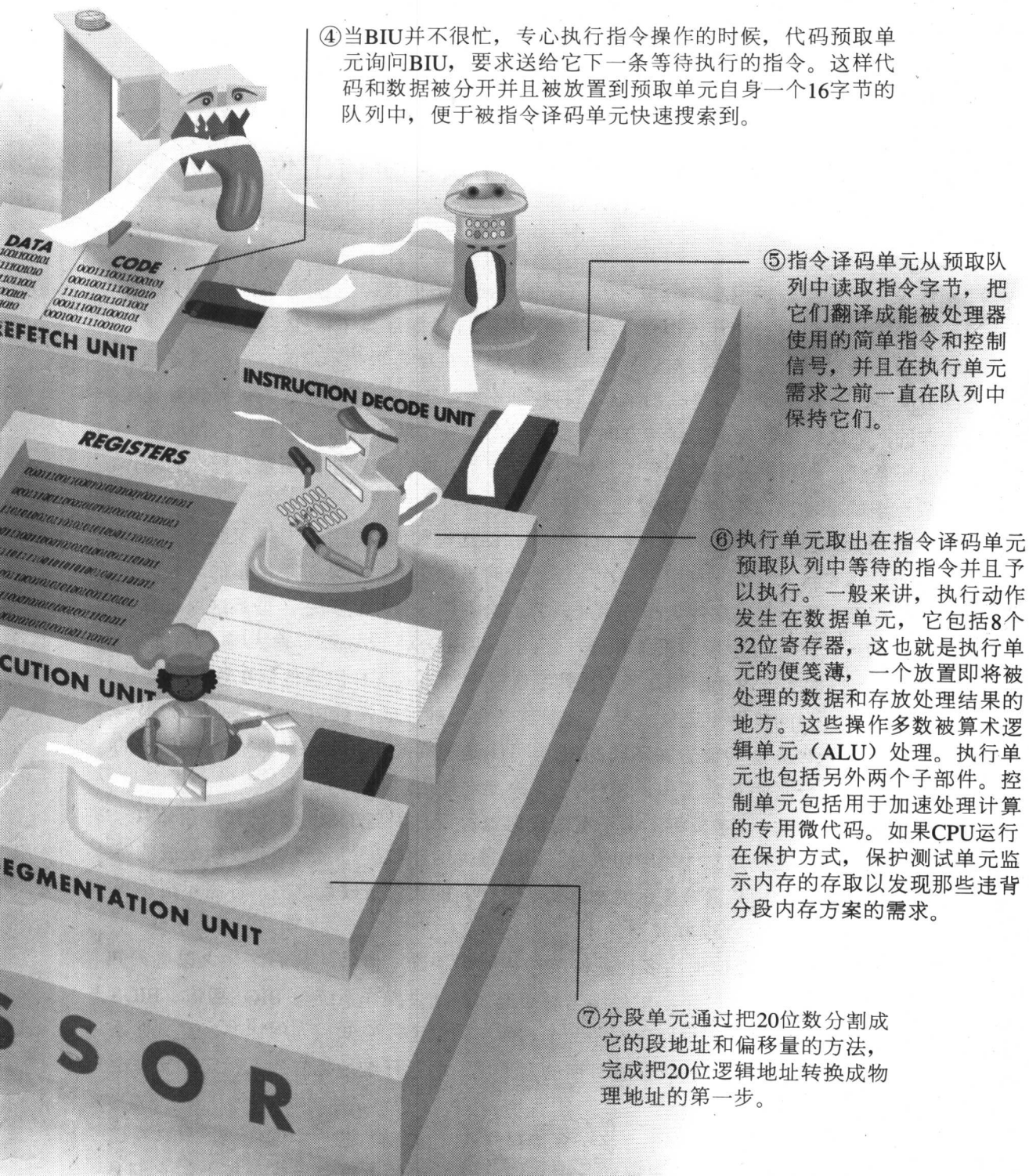
①当你运行一个程序的时候，PC机的操作系统将永久存储在硬盘上的程序，拷贝一份到RAM中。

③数据和代码被总线接口单元（BIU）从RAM中搜索出，并把段地址和偏移量转换成线性地址，所有从微处理器经过的数据和代码也必须经过BIU才能到达RAM。

⑨通过称作流水线的处理，被多个部件完成的全部操作却可能是同时发生的；流水线的目标就是保证处理器的多个部件用尽量短的时间等待下一个数据或指令。

⑧分页单元完成把逻辑地址转换成物理地址的最后一步，就是对它进行适当的调整，当处理器在使用称作页面的部分存储器时，这种调整可能是很有必要的。该单元产生一个实际信号，并把它放在微处理器地址线的管脚。分页单元把地址传送给BIU，BIU使用地址把从执行部件中取得的数据结果放到物理存储器中的正确位置中。







第二章 BIOS如何与软件协同工作

假设你坐进了一辆汽车的驾驶室却发现那儿没有方向盘，它被飞机的操纵杆替代了，再者，你对面的一个开关取代了脚刹闸，那么你能只能靠鼻子操作了。

幸运的是，你永远不会面对这样的问题，汽车的机械装置是不同的，有些具备强有力的方向盘，前轮驱动，和盘式刹车，而有些则不然，但你无需知道这些机械装置是如何工作的，你仅仅需要知道怎样转方向盘和踩刹车，在中间的一层，控制杆、传动装置以及液压系统把你和机械装置分开了。你只需专心地开车，这些机械装置会把你的动作翻译成汽车的运动，改变方向或减速。

你的个人计算机也是一样的，它也有一层东西把你和你的应用软件与又脏又苦的硬件工作分开。实际上，这儿有两个层次。无论何时当你在 C：提示符下做事的时候，都会遇到其中第一层，这一层叫做操作系统，我们会在下一章中讲到它，但是这里还有一个更基本的层次，它位于操作系统和硬件之间，即BIOS，或称基本输入/输出系统。

BIOS使得你的应用程序和众多的操作系统以一种方式工作，当他们在兼容机上运行时，就象在IBM的机器上运行一样。它把软件命令翻译成常规PC机需要执行的那些命令信号，由于硬件层的信号在不同的PC上可能是不同的，为确保软件在多种品牌的IBM兼容机上运行时都正常工作，BIOS起了决定性作用。

如果没有BIOS，程序员将不得不重写针对每种硬件配置的操作系统和应用软件。操作系统必须知道读或写文件的确切信号，识别键盘的动作，为上千个元件组装成的特定的驱动器、键盘、监视器等显示文本。购买一个新的大容量硬盘之后，你将不得不更换字处理软件版本，来保存其文件。

BIOS掌握了发送和识别与多种硬件部件相关的正确电信号的核心，代之以必须知道某个驱动器怎样工作。一个应用程序只需知道如何通过操作系统与BIOS通信。BIOS认真对待应用程序的请求，就好象有了一个高效率的助手，它知道如何让办公室文件系统工作。你所需做的就是将文件放在外面的蓝子里，助手仔细地把它放入指定的文件柜中，指定的抽屉当中去。

BIOS工作部分是安排在一个或多个只读存储器（ROM）芯片中的代码，只读存储器是一种不能被修改的存储器类型。这些芯片通常在PC的主电路板或母板上，它们是