北京希望电脑公司DOS技术丛书

尚未公开的DOS秘密

—MS—DOS中内部保留的功能调用和数据结构

吴 双 编译
许志平 审校

海洋出版社

北京希望电脑公司DOS技术丛书

尚未公开的DOS秘密

——MS-DOS中内部保留的功能调用和数据结构

吴　双　编译

许志平　审校

海洋出版社

内 容 提 要

本书收集了关于MS—DOS内部功能和数据结构的各种资料，对它们进行了分析，归并和整理。首次发表了以前从未公开的MS-DOS内部功能和数据结构，并介绍了如何在程序设计中运用这些知识。

本书首先讨论了未公开功能及数据结构的作用和用法。然后以此为基础，详细介绍了MS—DOS资源管理，文件系统，驻留程序设计方法，多任务模拟，命令解释器以及调试程序的编写等专题，从各个侧面展开了MS—DOS的内核。最后，逐项引出并解释了到目前为止能收集到的所有未公开的MS—DOS功能调用和数据结构。

本书适用于在MS—DOS上进行软件开发的程序员，也适用于研究MS—DOS功能和结构的研究人员。

欲购本书的用户可直接与北京8721信箱联系，

邮码：100080，电话：2562329

* * * * * * * * * * *

尚未公开的DOS秘密

吴 双 编译

许志平 审校

阎世尊 责任编辑

* * * * * * * * * * *

海洋出版社出版（北京市复兴门外大街1号）

双青印刷厂印刷

开本：787×1092 1/16 印张：28.625 字数：685千字

1991年5月第一版 1991年5月第一次印刷

印数 1—3000册

ISBN7—5027—2138—X/TP·30

定价：17.00元

前　言

自从MS-DOS进入市场以后，大约有100多个功能调用被列为“保留的”。虽然没有任何公开资料介绍它们。但是，在驻留程序，多任务，网络软件，可安装的文件系统，调试程序的设计，DOS扩展器的保护方式以及WINDOWS 3.0等各方面的应用中，都大量地使用了这些未公开的功能调用。作为一个MS-DOS程序员，我们有必要对这些未公开的功能进行研究，并掌握使用它们的技巧。

本书以此为目的，充分地揭示了DOS中各种保留的功能和内部数据结构。本书不仅介绍了如何使用未公开的功能的一般性知识，还介绍了使用它们时要注意的问题。在这个基础上，详尽地介绍了MS-DOS的资源管理，文件系统，驻留程序，多任务环境，命令解释器，DOS外壳以及调试程序等各方面的内容。本书还提供了一个称为 INTRSPY 的程序来显示MS-DOS的内部动作。最后，本书列出了所有未公开的功能的清单。

本书的英文作者们都是在DOS上具有丰富经验的专家，其中甚至包括MS-DOS本身的设计者。他们通过介绍DOS的各个专题，为我们逐步展开了MS-DOS的内部细节。

本书对于使用MS-DOS或在 MS-DOS 上进行软件开发的高级程序员来说是一本非常有用的参考书。其中的一些知识不仅是相当重要的，而且也是第一次公开发表的。

吴　双

目　录

引　言

Andrew Schulman

我先讲一个与本书有关的故事。

几个月以前，我的一个同事自以为我是个DOS高手，请我编写一个取消MS-DOS的L驱动器的程序。虽然我不清楚这个程序的目的，但肯定是有些用户对如何从内存中删除Microsoft的CD—ROM扩展功能的方法不满意，想用这个程序执行这方面的任务。我试验的各种不同的方法，都没有奏效。其中包括MS—DOS的取消设备重定向功能（DOS功能5FH的子功能04H）。

后来，我查阅了Ralf Brown的中断列表一书。其中介绍了一个称为"取表之表地址"的DOS功能（功能52H），它被标记为"DOS 2.0以上内部使用"。这个功能从未在任何DOS正式资料中介绍过。IMB DOS 3.3的技术手册在介绍完功能4FH后就直接介绍功能54H。甚至Ray Duncan的《MS—DOS高级程序设计》一书也把功能52H列为"保留的"。但是，正是这个未公开的DOS功能可以解决取消驱动器L的问题。一旦了解了功能52H，编写程序就易如反掌了；反之，没有这个信息时，一切都无从下手。

讲这个故事并不是要介绍编写某个程序的具体细节。我的主要目的是要指出，确实有一些应用程序中需要使用那些没有在Microsoft或IBM资料中介绍过的DOS功能。

当然，做为一个DOS程序员，我过去已经注意到确实存在很多来公开的DOS功能。在许多计算机杂志中或电子信息板中可以找到关于这类问题的各种介绍。但对于我来说，看到这些内容分散在很多不同的资料中，促使我们开始编写这本书。这里，我们把很多分散的，不清晰的或近乎矛盾的说法统一起来，并且考虑DOS版本的差异带来的变化。很明显，这样一本集中所有未公开的DOS功能和数据结构的书是非常必要的，更何况它还介绍了不同DOS版本之间的差异以及如何安全地应用这些未公开的功能。

我们集中了一批对这方面有独到研究的软件工程师。Jim Kyle曾经在著名的《DOS程序员参考手册》提供了未公开功能的资料。Ray Michels在MS—DOS的论文中论述过未公开的DOS功能。Ralf Brown曾经清楚地列出了所有DOS功能的中断表。Tim Paterson不仅描述了未公开功能的主要性，而且为全书进行了技术把关，他本人就参加编写了MS-DOS系统。另外，为了帮助读者深入地了解DOS功能，David Maxey编写了INTRSPY程序。

0.1　本书的内容

很多研究过DOS技术手册的程序员都会发现一个现象。手册中列出的功能号经常被标记为"保留的"，或者根本就不提某些功能。本书的任务就是补齐并详细地解释这些没有提到的DOS功能。

除此之外，本书还详细介绍了DOS中未公开的数据结构，如内存控制块（MCB），当前目录结构（CDS），数据交换区（SDA）以及表之表（List of Lists）。书中还详细介绍了某些公开的数据结构中未公开的项目，如程序段前缀（PSP），文件控制块（FCB），驱动器参数块(DPB)，BIOS参数块（BPB）等等。这些信息都可以应用到实际程序中。我们还会详细介绍不同的DOS版本中这些数据结构的差异。

仅仅列出未公开的功能和数据结构是不够的，本书的主要目的是要提供使用这些内容的技术。其中一些内容属于DOS程序设计中的口头流传的技巧，它们在这里还是第一次公开印出。在本书中，我们提供的技术主要包括以下几个方面：

●访问主环境块

●遍历DOS内存链

●从命令行装入设备驱动程序

●用网络重定向器建立逻辑驱动器

●用中断2FH的功能AEH增加新的内部命令

●利用DOS数据交换区编写TSR

对于每一种具体任务，我们都试图提供多种不同的技术方案。这样做的目的有两个：一是比较各种方案的优缺点，二是用多种方式来安全地使用未公开的DOS功能。

本书中的程序已经在MS—DOS和PC—DOS中仔细测试过，具体测试的版本是DOS 2.x，3.xx和4.xx。这些程序还在OS/2 1.1和2.0中的DOS兼容环境中测试过，同时也在DR—DOS中测试过。令我们惊奇的是，尽管这些程序使用的是未公开的DOS功能，但它们却能在相当多的DOS版本或伪DOS版本下正常工作。这样看来，未公开的DOS功能并不是十分可怕，同时也可以证明本书对不同DOS版本之间的差异做了大量工作。

0.2 书中的内容是保密的吗？

本书中没有任何内容是绝对的秘密。你可以在计算机杂志或电子信息板上找到各种形式的关于未公开功能的说明。本书与众不同的就是把散见于各处的资料组合在一起，并且提供了大量的程序来演示如何使用这些资料。

本书中的作者并不打算破坏与Microsoft之间的任何非扩散约定，而且本书中的内容也不在任何非扩散约定的限制之内。

虽然讨论未公开DOS这件事事实上是公开某些秘密，但既使是Microsoft本身的杂志中也常常讨论未公开的DOS功能。因此，也就没有理由限制本书中的讨论。

另外，本书中包含的某些资料也可以在其它场合中得到。网络重定向接口（中断2FH的功能11H）也是Microsoft本身公开的。本书第一次公开的内容包括DOS数据交换区，可安装的命令接口以及用DOS功能4CH删除内存驻留程序。

0.3 为什么要使用未公开的DOS？

所谓未公开的功能和数据结构就是Microsoft或IBM资料中未提及的或标记为“保留的”内容。

例如，中断21H中保留的功能50H到53H属于DOS的一部分，它们应该算做未公开的DOS功能。但对于DOS网络重定向器，MS-Windonws，PC LAN来说，情况就比较复杂了。因为这些内容本身的界限就不是很清楚。我们这里所说的未公开的DOS内容是指一个比较狭窄的范围，它并不包括那些难以得到的资料。例如，中断34H到中断3EH是Microsoft用来处理浮点运算模拟的，而中断3FH是Microsoft用来处理覆盖程序的。这些内容都不属于本书的范围。

另一方面，本书中不仅包含未公开的DOS功能，还介绍了那些公开的DOS功能中未公开的部分（如DOS功能4BH的子功能01H）以及公开的数据结构中的未公开的部分(如FCB)。

除此之外，本书还提及了某些功能中未公开的副作用（如中断21H的功能13H）。

0.4 潘多拉的盒子

这里要对未公开的DOS功能的使用提出一些相反的看法。本书的所有作者都认为只有在公开的DOS接口满足不了我们的需要时才可以使用未公开的内容。我们要提醒读者，之所以把它们称为未公开的功能，就是为了要尽量避免使用它们。

我们编写本书的目的是要介绍一个DOS程序设计的新世界，同时提供关于未公开的DOS的统一的和可信赖的资料。但是，在我们面前仿佛有一只潘多拉的盒子。如果这本书导致出现一大批使用未公开DOS功能的程序，一旦Microsoft对DOS进行了改进，这批程序就必须要经过重新测试。

但话又说回来，使用未公开的DOS功能的程序已经为数不少。这就强迫Microsoft在OS/2的DOS环境中也不得不照原样复制所有未公开的DOS功能。

实际上，这个问题的核心在于如何看待计算机系统的信息隐藏原则。根据信息隐藏原则，计算机操作系统要提供编写程序所需要的所有内容，而程序员也要保证不去查看接口后面隐藏的内容。这样，系统的内部改变并不会影响程序的运行。但问题是，MS—DOS并没有提供软件开发所需要的一切内容，这就迫使我们去研究接口后面隐藏的未公开的内容。

0.5 对读者的要求

对于那些已经熟悉DOS程序设计的读者来说，本书会提供相当有用的信息。很可能有些读者在没有完全了解标准DOS程序设计的协议以前，就急于通过本书了解未公开的DOS功能，对于这批读者，我们在第二章中介绍了使用DOS功能的基本方法。

如果读者熟悉C或汇编语言，将从本书中得到很大收益。对于那些不了解这两种语言的读者，我们在第二章中还提供了PASCAL和C的程序实例。

本书对读者的最低要求是他要熟悉IBM PC及其兼容机。非程序员的读者恐怕只能从第一章中了解一些感兴趣的内容或从第六章中了解一些DOS命令解释过程的内容。

0.6 作者简介

Ralf Brown：卡内基-梅隆大学计算机博士。

Jim kyle：编写过多种计算机方面的著作并发表过上百篇文章，参加过《MS—DOS大全》的编写。

David Maxey：有12年系统开发经验，编写了本书中的INTRSPY程序。

Ray Michels：一直在MS—DOS上做工作，编写过未公开的DOS方面的文章。

Tim Paterson：参加过MS—DOS 1.x的设计。

Andrew Schulman：编写过《Extending DOS》一书，并为多家计算机杂志撰稿。

第一章　关于未公开的DOS的用法

Andrew Schulman

在IBM PC，PS/2及其兼容机上的MS-DOS操作系统是目前世界上应用最应泛的操作系统。据估计，MS-DOS上商用的和非商用的应用程序总数已超过两万种。而安装了DOS系统的用户大约在3000万到5000万之间。虽然这种估计可能带有商业上的夸大，但比较保守的估计也表明有3000万个DOS在运行。这个数目大大超过了任何其它操作系统。

在这3000万台机器上，MS-DOS（或PC-DOS）不仅提供了我们所熟悉的用户接口A>，还提供了一个程序员接口。正如一般用户使用dir *.exe命令一样，使用DOS的程序员也通过中断21H来打开文件或申请内存。MS-DOS程序员接口由一组软中断组成，其中最重要的是中断21H。

MS-DOS的技术资料随着MS-DOS本身流传到世界的各个角落。作为一本DOS程序设计的圣经，Ray Duncan的《MS-DOS高级程序设计》一书提供了大量的有用信息。很多这类关于DOS程序设计的书在介绍输入输出，磁盘文件，内存分配以及错误处理等专题后，都会不约而同地列出一组中断21H功能调用。其中罗列了从功能0（程序终止）开始的一系列功能调用。

很明显，MS-DOS的世界由于这些书籍和资料的支撑显得非常清晰。同其它操作系统相比，MS-DOS是一个很小的操作系统，因此好的程序员可以把DOS程序设计技术彻底地掌握起来。

但是，情况并不总是这样。打开任何一本MS-DOS的程序设计手册，你都会发现中断21H的功能号从4 FH（找下一个文件）到54H(取验证标志）之间没有任何内容。既使是在Ray Duncan的书中，功能50H到53H也被标记为“保留的”。

在本书的附录中你会发现，这些MS-DOS程序员接口中没有提到的内容已经被详细地列出了：

中断21H功能50H——设置PSP

中断21H功能51H——取PSP

中断21H功能52H——取表之表

中断21H功能53H——转换BPB

一般DOS手册中另外一个未说明的内容是功能5DH，它包含12个子功能。尽管MS-DOS本身并不大，但它却包含很多没有完整地描述的部分。

既使中断21H中公开的功能中也有一些未公开的子功能。例如，功能4BH的未公开的子功能01H可以装入一个程序而并不执行它，它是编写DOS调试程序的核心。另外一些功能中的某些行为或副作用在一般手册中也没有讲透。

除了中断21H以外，DOS还占用了其它一些软中断。如中断2FH就是作为网络重定向器的程序员接口（中断2FH的功能11H）。另外DOS中实用程序APPEND.EXE也使用了中断2FH的功能B7H。

实际上，这些未被提及的功能仅仅是未公开的DOS功能的一部分。另外的内容包括一组

用的数据结构，如程序段前缀（PSP），驱动器参数块（DPB），表之表，内存控制块（MCB），系统文件表（SFT）等等。

1.1 为什么有些功能不公开

粗看起来，开发MS-DOS的人没有完全公开操作系统的所有功能是一个错误。既然让用户使用它，为什么又不详细地介绍它呢？

现实情况是，任何软件中的开发者都会保留一些特性不向外界公开。一旦在文档中说明了某种产品特性，那么在以后的版本中就要永远支持这个特性。实际上，Microsoft为了维持已经公开的各种MS-DOS特性曾吃了不少苦头。例如，它不得不兼容CP/M的文件控制块（FCB）以及DOS的程序段前缀（PSP）结构，其最大的失策就是过早地公布了这种内部结构。

Microsoft为维持这种向下兼容的功能做出了大量的努力。例如，在DOS 1.0的手册中指出，除向使用INT 21H以外，应用程序还可以用CALL 05H指令调用操作系统的功能。作为CP/M的后续者，DOS不得不承受很多重要的程序（如WordStar）已经使用了这种方法的现状。MS-DOS只好在PSP偏移量为5的位置放一条JUMP指令，来支持CALL 05H的方法。由于在PSP中公开了这些位置的作用，结果每个DOS程序（甚至包括运行在80486上的程序）。都带有CP/M时代64K内存打下的烙印。既然知道作出改动非常困难，公开所有的特性等于给自己找麻烦。

从Microsoft的观点看，不公开任何DOS的区域是最妥善的选择。它声明，任何使用自己发现的功能和数据结构的程序可能在未来的版本中不会正常工作。Microsoft为此专门在1988年发布了一个文档（号码为Q34761）。它声称："Microsoft不提供任何有关未公开的特性的信息。之所以不公开它们，是因为我们不保证在今后的DOS版本中这些内容会继续存在"。

这个声明当然是合理的。本章的目的是为了向程序员提供未公开的DOS特性并解释这些内容为什么会使DOS显得如此神秘。我们还介绍了很多重要的商用软件是如何使用未公开的DOS特性的。

1.2 未公开功能的重要性

为什么我们对未公开的DOS特性如此关心呢？难道只是为了使INT 21H的功能号变得连续起来吗？

当然，其中的原因之一纯粹就是好奇心。一看到某个功能被标记为保留的，我们总会提出这样的问题：为什么要保留？为谁保留？

实际上，好奇并不是研究未公开DOS特性的主要原因。很多CPU中也都有一些保留的位不需要我们了解。本章的后面将要指出，使用未公开的DOS特性是与使用未公开的硬件特性不同的。这里我们先集中考虑未公开的DOS特性。

研究未公开DOS特性的真正原因来自于MS-DOS本身的重要性。前面已经提到，全球共有3000万台机器正在运行着MS-DOS。Microsoft曾经想用更新的OS/2来补充MS-DOS，但是MS-DOS仍然没有受到任何冲击。

表面看来，MS-DOS实在没有什么更多的高深内容，因为它的代码非常短小。组成DOS核心的两个文件（IO.SYS和MSDOS.SYS）与命令解释器COMMAND.COM加在一起也不

是110K大小。这么小的程序是如何引起整个工业界（各种书籍，杂志，电子信息板以及用户团体）的重视呢？MS-DOS所有的源程序量加起来还没有本书的字数多。

问题的关键是DOS的可扩展性。如此小的DOS完全支持各种扩展，而不是限制这些扩展。DOS提供了一组新的功能来支持各个方向的扩展。这一点是DOS获得巨大市场的关键，也是DOS最成功的地方。

到底DOS可以进行哪些方面的扩展呢？除了大家都熟知的内存驻留程序（TSR）以外，还可以列出以下几个方面：

- ●Windows系统
- ●多任务系统
- ●网络支持
- ●可安装的文件系统
- ●调试器
- ●保护方式下的DOS扩展器

为了把这种获得巨大成功的操作系统在某些方面进行扩充，我们当然需要了解那些应该了解的内容，其中也包括未公开的特性。

1.3 允许使用，但不保证

事实上，很多未公开的DOS功能和数据结构正是扩充操作系统的关键。虽然我们说DOS允许进行无限制的扩充，但我们并未说过DOS会支持这种扩充。之所以这样讲，是因为DOS把所有扩充能力都隐藏在未公开的部分里。

1.3.1 不支持TSR

举例来说，DOS虽然允许驻留程序的存在，但并不支持它。在MS-DOS中，可以把程序驻留在内存并用它安装相应的中断处理程序。DOS为此提供了三个公开的功能。其中功能25H用来设置中断向量，功能35H用来得到中断向量，功能31H用来使程序终止并驻留。这里没有任何障碍阻止你替换INT 21H本身。

但是，DOS也没有对这方面的工作给出进一步的支持。事实上，要想把你的程序真正嵌入到DOS中还需要很多未公开的功能。其中包括中断21H的功能34H，50H和51H，以及中断28H。

把对TSR的支持限制在MS-DOS未公开领域已经是众所周知的了。1986年，Microsoft和其它一些公司的代表在一起商谈是否要建立TSR的工业标准，其中一个主题就是讨论如何对待DOS未公开的特性。根据Microsoft当时的记载，Borland和Lotus都曾提出，某些未公开的DOS特性对于建立彼此相容的TSR是非常关键的，其中包括很多DOS内部标志和系统调用。

根据本书的第五章，我们可以了解到，为了保证TSR的一致性，下面的DOS功能是非常重要的：

- ●中断21H功能34H（返回InDOS指针）
- ●中断21H功能50H（设置PSP）
- ●中断21H功能51H（取得PSP）
- ●中断21H功能5D06H和5D0BH（取DOS数据交换区）

●中断21H功能5D0AH（设置扩展错误信息）

●中断28H（键盘忙循环）

到目前为止，Microsoft仍然没把这些信息加在MS-DOS程序员接口中。从DOS3.0起对功能51H的限制已经没有必要了，MS-DOS提供了一个等价的功能调用（功能62H）。

实际上，Microsoft也曾经提到过一些支持TSR的未公开特性。它出版的《DOS百科全书》中就提到过上面列出的一些功能。但是，这些内容在该书的参考手册部分仍然被砍掉了。而且其中的功能34H还附加上Microsoft不保证其正确性的注解。现在，DOS中支持TSR的未公开信息已经被相当多的人所了解。为了编写正确的和稳定的TSR，在使用未公开功能这一点上，人们已经有了共识：使用未公开的功能在某些情况下不仅不会产生不可靠的软件，相反还是编写正确软件的必由之路。

1.3.2 网络重定向器

DOS中另外一个允许扩展但不支持扩展的领域是DOS文件系统。任何在网络上使用过PC机的用户都知道，可以把另一台机器的磁盘像使用自己机器的磁盘一样使用。例如，你可以通过DOS命令DIR E:命令来查看另一台机器（甚至可以是Macintosh机器）上的文件目录。这种结果是如何得到的呢？DOS是如何从另外一台机器上得到目录的呢？你自己是否可以编写出这种软件呢？种种问题不一而足。

另外一个令人注目的例子是Microsoft的CD-ROM扩展（MSCDEX），这是一个利用未公开的DOS功能把CD-ROM转化成标准DOS设备的例子。很明显，DOS中一定有一些功能能够把CD-ROM（光盘）上的文件系统转化为标准DOS设备上的FAT结构的文件系统。

Microsoft曾经透露出一些实施这种转换的信息。Microsoft发言人曾在一篇文章中指出，CD-ROM使用的驱动器不是本地驱动器，而是网络上的远地驱动器。根据这一说法，MSCDEX利用了MS-DOS中网络重定向器的功能。Microsoft对这部分内容守口如瓶，但本书第四章中详细介绍了如何通过中断2FH的功能11H来利用网络重定向器功能。

这样看来，使用未公开的DOS功能是完全必要的。Novell曾经在Microsoft增加网络重定向器功能以前利用扩充中断21H建立了高性能的网络系统。虽然它没有使用中断2FH的功能11H，避免了未公开的网络重定向器功能。但在它的软件NetWare中也还是利用了其它未公开的DOS特性。

1.3.3 对调试器的支持

当你要编写DOS下的调试器时，就会用到装入程序但不执行的功能调用。DOS的中断21H功能4BH的子功能01H提供了这一功能，它也被用在DEBUG，Symdeb，CodeView和Turbo Debugger中。不幸的是，MS-DOS技术手册中在介绍功能4BH时只提到了子功能00H和03H，而没有提到子功能01H。

1.4 对于未公开部分的恐惧

我们已经看到，DOS中包含了大量的未公开的内容。Microsoft保留这些部分是为了能够在以后的DOS版本中不受限制地改动它们。通过阅读本书你会了解所有这些神秘的功能和数据结构。虽然你能通过INT21H的功能52H返回DOS内部的变量表，或者用功能4BH

的子功能01H装入一个程序，但如何把这些知识用在实际的程序设计实践中呢？

当然，Microsoft是不主张这样做的。使用这种未公开的特性可能导致产生不稳定的和不可移植的软件。一般来说，使用未公开的特性是不符合软件工程的规范的。但是，它们却能帮助我们编写出正确的TSR，网络驱动程序和调试器。

在未公开的DOS内容周围似乎总有一种神秘的气氛。一些程序员尽量避免使用它们，尤其是某些书籍的作者总是提出类似Microsoft的警告，同时避免在正式场合提到这些未公开的功能。

但是，使用未公开的DOS功能并不一定限制程序的适用范围。只要按照本书给出的技术要求去做，就可以在程序中安全地使用各种未公开的特性。事实上，很多成功的商业软件也利用了未公开的DOS特性。前面我们已经介绍过的Lotus和Borland两家公司也都使用过这些内容。造成这种现象的原因是，大型软件工厂冒险使用未公开的DOS功能会得到更大的好处。当然，这类程序必须经过更充分的测试。

从某种意义上来说，使用未公开的DOS功能与在程序中使用GOTO语句的情况很相似。虽然我们应该尽量避免使用它们，但必要时还得依赖它们。

1.5 80X86中未公开的内容

在计算机的其它领域中也有很多被标记为保留的特性。下面让我们观察一个例子：Intel处理机中的保留内容。

在Intel 8088的资料中，INT 05H被标记为保留的。IBM用8088构成PC机时把INT 05H设计成ROM-BIOS中打印屏幕的功能。当Intel推出80286时，它把INT 05H从保留的状态变为处理越界中断。这样，INT 05H的功能在Intel和IBM之间就有了很大的分歧。其原因就是IBM使用了Intel保留的中断号码。

IBM当然可以完全不在乎INT 05H原来的功能，那么我们也可以把DOS功能52H的功能用在我们的程序中吗？事实上，确实有一些商业软件就是这样做的。

1.5.1 汇编语言中未公开的部分

在汇编语言中，AAD和AAM指令有一些未公开的效果。它们不仅可以用来乘以或除以10，也可以有其它的用处。使用这些指令的未公开特性当然是非常危险的，但使用DOS中未公开的特性则有所不同。

首先，能够生产80x86芯片的厂家不只Intel一家，NEC，AMD和Harris不一定保证实现这些未公开的特性。而另一方面，生产DOS的厂家只有Microsoft一家，而且Microsoft也要求它的OEM厂家在修改DOS3.0时不能改变其内部结构。

其次，80x86的覆盖面要比DOS的覆盖面更大。DOS 3.x和DOS 4.x之间的差别相对来说要比80286和80386之间的差别更小。

第三，Intel在对待未公开的特性这方面比Microsoft更强硬。虽然Microsoft没有公开支持这些功能，但它为了保证重要的PC商用软件的运转，还是保留了DOS中的兼容性。在关于未公开的DOS功能的核心部分，Microsoft并没有做出什么改动。

第四，利用CPU的未公开特性的程序要比利用DOS中未公开特性的程序少得多。前者带来的好处只是提高了效率，而后者却帮助解决了原来不可能解决的问题。

最后，除了多任务环境和调试程序不得不利用取PSP段地址和置PSP段地址两个功能

外，大多数程序还是尽量避免使用未公开的DOS特性的。此外，在使用它们时还进行了严格的版本检查。

1.6 使用了未公开的DOS功能的程序

我们需要了解的是到底有哪些商用软件使用了未公开的DOS功能。前面我们已经提到S过MCDEX，DEBUG，Symdeb和CodeView，下面让我们仔细地研究一下这些程序。

如果我们有这些程序的源代码，当然可以找到它们到底使用了哪些公开的或未公开的功能。如果通过反汇编这些程序来达到目的，一是会遇到授权问题，二是这种工作实在太枯燥乏味了。

前面我们说过，MS-DOS允许你替换系统中断，甚至包括INT 21H本身。根据这种想法，我们可以监视INT 21H（也包括其他DOS中断）来发现某个程序具体使用了哪些DOS功能。本书作者之一编写了一个称为INTRSPY的程序来完成这个工作。它是一个事件驱动和文本驱动的DOS调试程序。当然，还可以把它用于其它方面的工作。本书的第八章将详细介绍这个程序的功能。你可以为INTRSPY程序编写出一个描述文件来记录程序是如何使用未公开的DOS功能的。最简单的描述文件并没有利用上INTRSPY的所有功能，它只监控何时使用了未公开的DOS调用。

```
; UNDOC.SCR (abridged version)
intercept 21h
    function 1fh on_exit output "211f: Get Default DPB: " DS ":" BX
    function 32h on_entry output "2132: Get DPB: " DL
    function 34h on_exit output "2134: InDOS flag: " ES ":" BX
    function 50h on_entry output "2150: Set PSP: " BX
    function 51h on_exit output "2151: Get PSP: " BX
    function 52h on_exit output "2152: Get List of Lists: " ES ":" BX
    function 53h on_exit output "2153: Translate BPB"
    function 55h on_entry output "2155: Create PSP: " DX
    function 5dh subfunction 06h
        on_exit output "215D06: Get DOSSWAP: " DS ":" SI
    function 60h on_entry
        output "2160: Canon File: " (DS:SI-byte,asciiz,64)
    function 4bh
        ; use this just to show which program made undoc DOS call
        subfunction 00h
            on_entry
                output (DS:DX-byte,asciiz,64)
        subfunction 01h
            on_entry
                output "214B01: EXEC debug: " (DS:DX-byte,asciiz,64)
    function 4ch on_entry output "------------------------------"
    function 31h on_entry output "---------- TSR ----------"
    function 25h
        on_entry
            if (al == 28h) output "SetVect INT 28h: KBD busy loop"
            ; not complete, because many programs unfortunately hook
            ; interrupts by poking the lowmemory interrupt vector table
intercept 2eh
    on_entry output "2E: Execute command"
```

装入INTRSPY以后，把描述文件UNDOC.SCR传送给它，然后可以运行一系列DOS下的程序。假定我们使用了DOS下的SUBST，JOIN，PRINT，CHKDSK和APPEND：

```
intrspy
cmdspy compile undoc.scr
subst d: c:\swap
\dos33\join a: c:\floppy
print
chkdsk
append \undoc\intrspy
cmdspy report undoc.log
```

在执行完这些命令之后，所有使用未公开DOS功能的动作就会记录在一个称为UNDOC.LOG的文件中。

```
------------------------
 C:\DOS33\SUBST.EXE
2152: Get List of Lists: 028E:0026
------------------------
 C:\dos33\JOIN.EXE
2152: Get List of Lists: 028E:0026
2152: Get List of Lists: 028E:0026
2152: Get List of Lists: 028E:0026
------------------------
 C:\DOS33\PRINT.COM
2151: Get PSP: 1376
2150: Set PSP: 1376
2152: Get List of Lists: 028E:0026
SetVect INT 28h: KBD busy loop
2134: InDOS flag: 028E:02CF
2150: Set PSP: 1376
2151: Get PSP: 1376
2150: Set PSP: 1376
2150: Set PSP: 1376
--------- TSR -----------
 C:\DOS33\CHKDSK.COM
2160: Canon File: C:\
2132: Get DPB: 03
2160: Canon File: C:\FLOPPY
------------------------
 C:\DOS33\APPEND.EXE
--------- TSR -----------
 \undoc\maxey\CMDSPY.EXE
```

上面的输出结果显示出SUBST和JOIN都调用了INT 21H的功能52H，其中JOIN调用了三次这个功能。程序PRINT不仅使用了功能52H，还使用了功能50H，51H和34H，并且替换了INT28H。通过这些手段，PRINT实现了模拟多任务的后台方式。另外，CHKDSK除了使用功能60H以外，还使用了功能32H。而APPEND看上去没有调用任何未公开的DOS功能，但实际上，它所调用的中断2FH功能AEH由于没有在描述文件中出现，因而没有在输出结果中显示出来。

上面给出的描述文件仅仅列出了一部分未公开的DOS功能。实际上，很多DOS上的软

件还使用了其它一些未公开的功能。

1.6.1 Microsoft的其它软件

下面我们来观察一下Microsoft的其它软件：Windows 3.0，CodeView和MSC 6.0的程序员工作台（PWB）。根据前面装入的描述文件，我们可以把这些程序的运行结果显示出来：

```
\win30\system\win /e
\c600\bin\cv \undoc\mem
\c600\bin\pwb \undoc\mem.c
cmdspy report undoc.log

------------------------
C:\WIN30\WIN.COM
C:\WIN30\system\win386.exe
2152: Get List of Lists: 028E:0026
2151: Get PSP: 40A3
2150: Set PSP: 40A3
2150: Set PSP: 40A3
2134: InDOS flag: 028E:02CF
2152: Get List of Lists: 028E:0026
2151: Get PSP: 40A3
215D06: Get DOSSWAP: 028E:02CE
 C:\win30\system\KRNL386.EXE
2134: InDOS flag: 028E:02CF
2151: Get PSP: 4215
2150: Set PSP: 40A3
2150: Set PSP: 4215
```

```
 ------------------------
 c:\c600\bin\CV.EXE
2152: Get List of Lists: 028E:0026
2151: Get PSP: 40A3
2150: Set PSP: 0000
2150: Set PSP: FFFF
2150: Set PSP: 40A3
2150: Set PSP: 0E7B
2151: Get PSP: 0E7B
214801: EXEC debug: C:\UNDOC\mem.EXE
2151: Get PSP: 441D
2150: Set PSP: 0E7B
    .
    .
------------------------
 c:\c600\bin\PWB.COM
 c:\c600\bin\pwbed.EXE
    .
```