

OS/2 程序員指南

北京科海高技术集团公司培训中心

信通集团公司

一九八八年七月

OS/2 程序员指南

(用户培训资料)

范植华 施振川 尹彦芝
张国泸 杨红青 等译

北京科海高技术集团公司培训中心

信通集团公司

一九八八年七月

译 者 的 话

OS/2是英文Operating System/Two (操作系统/2) 的缩写, 代表运行在英特尔80286微处理器上的一个高性能的个人计算机操作系统。

英特尔80286是一种最新的微处理器。诸如PC/AT、XT/286, 以及PS/2 50、60和80型等最先进的微型计算机系统, 正是得益于80286高级的硬件功能, 才从它们的前驱PC、PC/XT、便携式PC和可转换PC, 以及PS/2 25和30中脱颖而出, 后者立足于第四代微处理器808X序列。

当以英特尔80286作为硬件基础的那些最先进的微型计算机系统, 再装配上OS/2这一最新软件产品时, 它们必将如虎添翼, 获得诸如可扩充的虚拟存储管理、多任务运行与高效调度、输入/输出保护一类高级功能和工具, 这些高级功能和工具通常仅在成熟的大型操作系统中才能看到。

OS/2分标准版与扩展版两种, 每种又分1.0和1.1两种版本。作为程序员指南, 本书专门介绍OS/2标准版1.0, 这是所有其他OS/2版本共同的结构框架。

全书共分三大部份, 十八章, 另加四个附录。其中, 序言、第二、十一、十二章由范植华同志翻译, 第一、三至十、十三章由刘兵、张海潮等同志翻译, 十四章由胡春庆同志翻译, 第十五章由张国沪同志翻译、第十六章由施振川同志翻译, 第十七、十八章由尹彦芝同志翻译, 附录A、B、C、D由杨红青同志翻译, 全书由范植华, 施振川同志审校。科海培训中心夏非彼同志担任编辑工作。

本书由于时间仓促, 错误在所难免, 敬请读者指正。

译者

1988.7.20

序 言

1985年秋, IBM公司和Microsoft(微机软件)公司宣布一项联合开发软件的协议, 根据这项协议, 签署为个人计算机研制新一代系统软件的合同。在以后的几年里, 这两个公司的一组设计人员和程序编制人员为了一个共同的目标——建立未来的PC应用程序设计基地, 而团结奋斗。

产品有许多目的, 其中, 有些一致, 有些相悖。但是, 一个至关重要的因素是清楚的: 新系统必须从今天系统软件的最佳性能入手, 根据这些最佳性能, 到达未来发展的

这在很多方面是一个挑战。起先我们没有意识到, 要在功能、合用性和性能之间作好权衡会如此艰难, 因为它们极度琐碎, 不易决断。几乎不存在一种选择能满足所有用户的需要。

在此期间, 对于新的系统作过种种思考。伴随时间的流逝, 这些思考日臻成熟, 直至最终系统达到奇迹般的平衡, 满足所有人的所有要求。工业界众说纷纭, 传闻之中称系统为NEWDOS、ADOS、DOS 5.0、DOS/286以及CP/DOS等。

1987年4月2日, 在迈阿密海滩会议中心, 这一长久的企盼总算结束。IBM公司(以及那一天的晚些时候, Microsoft公司)宣告一种新操作系统的诞生, 把它命名为OS/2(即: Operating System/2)。OS/2是在包括佛罗里达州的博克·雷顿(IBM), 华盛顿州的列德蒙(Microsoft), 得克萨斯州的奥斯汀(IBM)和英格兰的苏斯列(IBM)在内的, 世界范围的开发实验室中, 一批天才的程序设计人员努力的结晶。

操作系统/2

实际上, OS/2不止一个系统。四月里的声明介绍了OS/2的两种不同版本: OS/2标准版与OS/2扩展版。OS/2标准版是IBM/Microsoft合作攻关的成果, 它分两种: OS/2标准版1.0是构成各种OS/2版本基础的服务和功能的核心集合, 它包括核心服务(比如存储、任务和中断管理)、I/O子系统(视频、键盘和鼠标)、实用程序, 以及一个基本的正文用户接口; OS/2标准版1.1则在上述基础之上增添全功能图像和窗口终端用户接口。

OS/2扩展版是OS/2标准版在通讯管理和数据库管理两个方面的延伸。扩展版的大部份设计和开发工作正由IBM在进行。它也分两种: OS/2扩展版1.0(以OS/2标准版1.0为基础)和OS/2扩展版1.1(以OS/2标准版1.1为基础)。

Microsoft公司也在市场上出售OS/2的版本(称作MS OS/2)。如同它们的IBM对应产品一样, 这些系统也是建立在共同研制的OS/2标准版系统的基础之上。

本书专述OS/2标准版1.0, 它规定了所有别的OS/2版本所采用的结构框架。你在这里所学的功课, 可以立即运用于OS/2和MS OS/2的其他版本。

系统构造

OS/2是一个新的操作系统, 明显地为英特尔80286/80386保护模式操作而构造。这一事实使得很多OS/2高级性能变成活生生的现实。OS/2戏剧性地背离了众所周知的PC DOS系统。借助于保护模式的硬件性能, OS/2突破PC DOS系统传统的局限性。极为引人注

目 录

序言

第一部分 介绍	(1)
第一章 为什么需要一个新的操作系统	(1)
1.1 PC DOS的局限性	(2)
1.2 OS/2环境	(7)
1.3 小结	(11)
第二章 英特尔体系结构	(12)
2.1 操作模式	(12)
2.2 存储模式	(15)
2.3 保护模式	(21)
2.4 OS/2的实现	(22)
2.5 小结	(24)
第二部分 OS/2如何工作	(25)
第三章 OS/2应用程序设计接口	(25)
3.1 API特性	(25)
3.2 API实现	(30)
3.3 小结	(38)
第四章 存储器管理	(40)
4.1 系统地址空间	(42)
4.2 应用地址空间	(43)
4.3 装入应用程序	(43)
4.4 存储分配和再分配	(44)
4.5 共享存储	(45)
4.6 段移动	(46)
4.7 段交换	(47)
4.8 请求装入	(49)
4.9 存储器部分分配	(50)
4.10 小结	(50)
第五章 多任务和动态链接	(52)
5.1 任务管理	(52)
5.2 OS/2任务模型	(53)
5.3 定时器服务	(60)
5.4 进程间通信	(61)
5.5 程序管理	(65)

5.6 小结.....	(70)
第六章 文件系统	(72)
6.1 文件系统模型	(74)
6.2 文件处理功能	(75)
6.3 实现	(78)
6.4 文件共享	(79)
6.5 卷管理	(83)
6.6 扇区缓冲	(85)
6.7 磁盘划分	(86)
6.8 设备I/O服务.....	(87)
6.9 输入/输出重定向	(88)
6.10 小结.....	(89)
第七章 资源管理	(90)
7.1 控制和响应.....	(90)
7.2 资源管理与API.....	(90)
7.3 设备管理.....	(91)
7.4 文件系统.....	(93)
7.5 处理器管理.....	(95)
7.6 存储管理.....	(96)
7.7 中断管理.....	(96)
7.8 小结.....	(97)
第八章 设备驱动程序	(98)
8.1 设备类型.....	(99)
8.2 设备驱动程序模型.....	(102)
8.3 设备驱动程序文件结构.....	(107)
8.4 内核申请包.....	(108)
8.5 设备辅助服务.....	(111)
8.6 兼容环境支持.....	(114)
8.7 OS/2 设备	(116)
8.8 字符设备监视程序.....	(116)
8.9 小结.....	(119)
第九章 I/O 子系统	(120)
9.1 子系统的必要性.....	(120)
9.2 OS/2 I/O 结构.....	(121)
9.3 显示器I/O子系统服务	(124)
9.4 键盘I/O子系统服务	(130)
9.5 鼠标I/O子系统服务	(131)
9.6 小结.....	(131)
第十章 会话管理	(132)

10.1 会话管理程序结构	(132)
10.2 系统外壳	(135)
10.3 命令解释器	(137)
10.4 小结	(151)
第十一章 系统专题	(153)
11.1 DOS 兼容性环境	(153)
11.2 错误处理	(160)
11.3 系统的可维修性	(165)
11.4 国家语言的支持	(166)
11.5 小结	(170)
第十二章 OS/2 的命令与实用程序	(171)
12.1 内部命令	(171)
12.2 命令解释器	(185)
12.3 系统实用程序	(186)
12.4 小结	(208)
第三部分 OS/2使用	(209)
第十三章 文件I/O服务	(209)
13.1 设备I/O	(209)
13.2 文件系统I/O	(225)
13.3 API族的讨论	(245)
13.4 小结	(246)
第十四章 I/O子系统	(247)
14.1 显示I/O	(247)
14.2 键盘I/O	(283)
14.3 小结	(295)
第十五章 存储管理	(297)
15.1 存储分配与释放	(297)
15.2 巨大存储	(301)
15.3 与其它进程共享存储器	(306)
15.4 子分配存储器	(310)
15.5 小结	(315)
第十六章 程序执行控制	(317)
16.1 建立进程	(317)
16.2 控制执行优先权	(325)
16.3 子进程管理	(330)
16.4 资源回收与系统信号	(331)
16.5 建立多个可执行的程序路线	(334)
16.6 临界区	(340)
16.7 小结	(341)

第十七章 高级多任务服务	(343)
17.1 OS/2信号量	(343)
17.2 OS/2时间服务的使用	(352)
17.3 进程间的通信.....	(360)
17.4 小结.....	(381)
第十八章 设备监控程序	(384)
18.1 监控程序组织.....	(384)
18.2 设备支持.....	(386)
18.3 使用监控程序.....	(390)
18.4 性能考虑.....	(401)
18.5 特殊考虑.....	(402)
18.6 小结.....	(402)
附录A OS/2功能调用	(403)
附录B API簇.....	(492)
附录C OS/2错误编码	(501)
附录D 链接控制语句	(516)
附录E 程序实例	()

第一部分 介 绍

第一章 为什么需要一个新的操作系统

在一九八一年秋季的一天，没有大加渲染，IBM宣布了为个人使用的微计算机——B M PC。随之而来的是个人计算机的一个迅猛发展时期。Microsoft（系统软件）、Lotus（商业应用软件）、Compaq（IBM PC兼容机）、BorLand（语言软件）以及Hayes（通信）等公司已数不胜数。不久以后，个人计算机这一年轻工业建立了两个标准件：PC机系统结构和PC DOS（MS DOS）操作系统。PC DOS成为大量应用软件赖以生存的基础。

PC DOS为应用软件提供了一个简单但强有力的编程环境。其基本功能包括：磁盘文件系统、命令解释器、批处理命令语言和一系列实用程序。不久，基本环境又有了扩展，其中包括全屏幕编辑软件、终端仿真程序和一系列新的商业应用软件。

1981年，基本的PC系统具有4.77MHz的8088微处理器、16K存储器和一个160K磁盘驱动器。一年以后，基本配置变成256K存储器和两个360K磁盘驱动器。1983年，PC XT将温切斯特硬盘技术带到PC机上。1984年，出现了具有80286微处理器的PC AT机。此外，1984年PC机首次得到了局部网络（LAN）的支持，多个PC机可以互连成一个单一系统。

在PC机硬件发展的同时，PC DOS也在发展。PC DOS的第一个版本支持最简单的PC硬件配置。随着新技术进入PC机，PC DOS也不断出现新版本。表1-1表示了PC DOS的发展过程。

表 1-1

PC DOS 的 版 本

版本号	发表时间	主要功能
1.0	1981	初始的操作系统
1.1	1982	支持双面（360K）磁盘
2.0	1983	支持PC XT机 • 10MB温盘 • 从硬盘自举 • 子目录和I/O重定向
2.1	1984	支持PCjr机
3.0	1984	支持PC AT机 • 20MB温盘

续表1-1

版本号	发表时间	主要功能
3.1	1985	• 新键盘 • 80286 (实模式工作) 支持PC Network (1.0)
3.2	1986	支持PC ConVertibe和PC Network (1.1) • 720K软盘 (3.5")
3.3	1987	支持PC/2机 • 微通道 • 1.44M软盘 (3.5") • 80386 (实模式工作)

随着技术向前发展, PC DOS的应用基础变得越来越复杂。集成软件已成为一个专用术语。所谓集成软件是一种应用程序, 它给用户一个完整的问题解决方法——决策支持(电子表格)、字处理、数据库和通信。然而, 这种软件带来存储器和磁盘的开销。PC DOS开始不适应这种应用要求了。

与此同时, PC的用户变得更复杂了。他们开始需求更昂贵的硬件投资。图形接口、窗口系统和多任务开始成为用户追求的目标。由于PC DOS扎根于早期的PC机配置, 因此它没有能力直接支持上述应用需求。事实上, 当PC AT于1984年问世时, PC DOS只能使先进的80286微处理器工作于实模式, PC AT只相当于一个速度快一些的PC。

但是, 计算机工业没有止步不前, 应用软件继续在复杂性方面增长。由于PC DOS不支持多任务, 应用软件的开发者只能建立自己的支撑环境。不久, POPUP式的应用软件开始产生。这些软件被装入到存储器中且被动地等待一个特殊键组合使之激活——被动的多任务。

但是, PC DOS的一个不容软件简单克服的缺点是640K存储器限制。随着集成软件和POPUP式软件的成长, 它们所需求的存储器容量也越来越大。真正的多任务和存储器被不可避免地缠绕在一起。

1987年4月2日, IBM和Microsoft宣布了OS/2, 新一代操作系统。PC DOS面向1981年的16K 8088 PC机, 而OS/2则面向新一代的80286/80386机器。本章将讨论OS/2如何克服PC DOS的局限和OS/2如何为下一代PC应用软件的建立铺平道路。

1.1 PC DOS的局限性

PC DOS是基于Intel公司微处理器的个人计算机的一个基本操作系统, 围绕着8088微处理器进行的设计。PC DOS提供了随机文件(硬盘和软盘)、设备(打印机和键盘)以及存储器管理等服务。PC DOS具有上千万个用户, 是当今世界上最流行的通用操作系统, 但它并不是不存在问题。

PC DOS最主要的局限来源于它所运行的CPU。即使当它运行在装备有更强功能的80286和80386微处理器的系统上时, PC DOS也是以8088模式工作。在本章中, 8088将意味着8088以及运行于8088模式的80286和80386的CPU。第二章将讨论这些微处理器如何

以其他模式工作。然而,从时间上考虑,将只从应用软件的工作环境的角度讨论这一问题。

1.1.1 存储器寻址能力

目前PC应用软件面临的最大挑战是PC DOS有限的存储器寻址能力。从理论上说,你可以在PC AT上安装足够的存储器(可达16MB),且你可以通过一些特定的程序使用它们,例如虚拟磁盘(VDISK)。然而,扩充了的存储器不能被用于运行一个PC DOS应用软件。PC DOS运行的CPU模式不能使程序访问1MB以上的存储器,因而操作系统和所有的应用软件都要生活在1MB的空间中。

1MB的寻址局限还应扣除大约360K的地址空间作为硬件I/O例程(BIOS),因而程序空间缩小到640K。在这640K中还要包括PC DOS本身(DOS 3.3占大约50K)以及它的一些扩展,因此留给应用软件的程序和数据空间不足590K。如果我们考虑一个700×200的电子表格,显然590K的空间是太小了。

这种情况还随着当你装入脱机打印程序(Print Spooler)和POPUP程序等多个“DOS扩展”软件时进一步复杂化。这些程序都要使用存储器且它们又必须都装入到590K的程序空间中。如果你已经使用了一种多任务的DOS外壳(Shell),例如TopView或MS Windows,你可能已熟悉590K的实际局限。尽管这些外壳声称使你可以在你的系统中运行一个以上的应用软件,但同时运行几个很有用的应用软件几乎是不可能的,原因就是存储器的有限。

1.1.2 静态的存储器管理策略

为什么应用软件要占据巨大的存储器空间呢?这个问题的根源是PC DOS的存储器管理策略。8088是一种简单的CPU,它没有提供任何有效的存储器管理功能来帮助操作系统。事实上,8088的程序是直接对PC的物理存储器进行寻址。因此,PC DOS不能“移动”应用软件的存储器。一旦投入运行,应用软件就拥有它所运行的物理存储器。这使得象POPUP式的应用软件和打印机脱机程序等可以作为操作系统的扩展部分留驻内存。这些程序在主要的应用软件启动前就被启动,且它们必须保持最大需要的存储空间,因为运行时不允许它们再申请新的空间。这种情况导致许多驻留程序占据了比它们实际需要多得多的存储器。

在某些情况下,更复杂的驻留程序要求你通过安装参数来定义该程序以后运行所需要的存储器资源。例如,在安装PC Network软件时要求用户提供一个往往使用户迷惑的数组参数,因为PC Network在启动后不能动态地分配存储器。因此,除非你计划运行在最小配置下,你必须告诉该程序你将使用多少资源(Sessions、names、bufilers等等)。

1.1.3 存储器完整性

在8088方式时,整个640K寻址空间表现为一个单一、连续的存储块,它不对操作系统代码和应用软件代码进行区分。任何程序可以检查和修改存储器的任何部分。

由于程序之间没有分离,因此一个应用软件中的一个简单程序错误可能导致整个系统的瘫痪。为了恢复一个“挂起”的应用程序你不得不多次关闭你的PC机电源。随着你增加并行运行的程序数,上述情况发生的概率也在增加。

在一个非存储器保护系统中由于程序的副作用而引起的错误实际上是不可诊断的。到系统失效时，已经太晚了。唯一的恢复措施就是重新启动系统。因此，一个实用的多任务系统要求至少将应用软件和系统存储器分开。

图1-1表示PC DOS的结构。请注意1MB地址空间是如何分配给BIOS、PC DOS、系统扩充部分和应用软件程序的。注意，应用软件可以偶尔修改PC DOS 或它的扩充部分。

1.1.4 I/O控制

从管理系统输入/输出 (I/O) 的角度看，PC DOS是一个“协作者”而不是一个“控制者”。然而，PC DOS并不是完全停滞不前的。8088模式给所有的程序完全的 I/O特权。这意味着任何应用软件可以不经过系统软件去直接修改硬件设备的状态。

更糟糕的是，在8088模式时，应用软件可以无限期地禁止系统中断。当中断被禁止时，外部事件（象定时器或通信数据的信号等）被丢失。例如，通信端口的最大速度是直接中断禁止的频度和持续时间有关的。如果你在多任务环境中禁止中断，则可能导致进程调度机制的失效，因为这种机制与时钟中断有关。当中断被禁止时，系统就不能运行另外的应用程序了。

这种情况的出现是由于PC DOS缺乏I/O控制机制，但它具有两面性，取决于看待这个问题的角度。一方面，它具有灵活性，导致许多流行软件建立在它的基础上，应用软件可以直接控制设备I/O。另一方面，由于大多数多任务操作系统均禁止应用软件直接控制 I/O以保证系统的完整性和坚固性，则PC DOS不可能成为这种环境的基础。

正如你在图1-1可以看到的，PC DOS的应用软件及PC DOS的扩展部分都可以跨过 PC DOS去直接访问硬件。

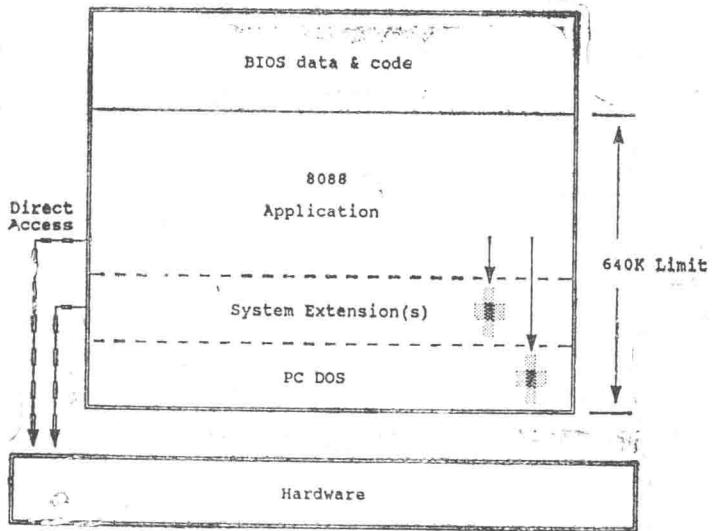


图 1-1 PC DOS系统结构

1.1.5 多任务

PC DOS是一个单任务的操作系统，在系统中只有一个应用软件运行。当加载一个应用软件时，PC DOS赋予它控制整个机器的权力。在很多情况下，应用软件需要多任务环

境，例如字处理软件要求在打印文件时可使用户继续文件编辑。由于PC DOS不提供多任务环境的服务，因此应用软件本身要承担建立多任务环境的责任。

表面上看，这是非常容易的。应用软件可以接收硬件定时器中断并调度它自己的任务。这种内部的调度器可以将CPU分配给不同的程序段使用。这也正是PC网络软件实现当应用软件在前台运行时服务器软件在后台运行的方法。

然而，如果应用软件所调度的任务要使用PC DOS的系统服务功能则事情就变得复杂起来。PC DOS是不可重入的——同时它只能服务一个请求。如果你试图在系统正为某个任务完成一种系统服务时再次请求PC DOS服务，则会导致PC DOS瘫痪。因此，实现自身多任务功能的程序通常要编制复杂的同步机制以确保PC DOS同时只接收一种服务请求。

作为一个真实的多任务系统，应允许I/O重叠，例如当系统从磁盘读数据的同时也可以向磁盘写数据。由于PC DOS存在重入性问题，因此即使应用软件自己设计多任务机制也不可能完全达到I/O重叠。

再考虑另一个问题。由于许多PC DOS的应用软件各自实现了专有的多任务机制，因此如果这些软件同时加载到一个系统中时非常可能产生严重的冲突，即相当多的这样的程序不可能与其它多任务程序共存。例如，你不能在Top View或MS Windows的环境下运行PC Network服务器程序，等等。

在PC DOS下实现多任务的另一个障碍是8088存储器管理策略。在8088模式时，CPU不能实现虚拟存储器管理，因此PC DOS不能在应用软件运行时进行存储器搬迁等功能。也就是说，PC DOS不能优化（或压缩）存储器。

1.1.6 系统扩展能力

PC DOS的应用软件编程接口（API）是基于软中断完成的。所谓软中断，实质上是

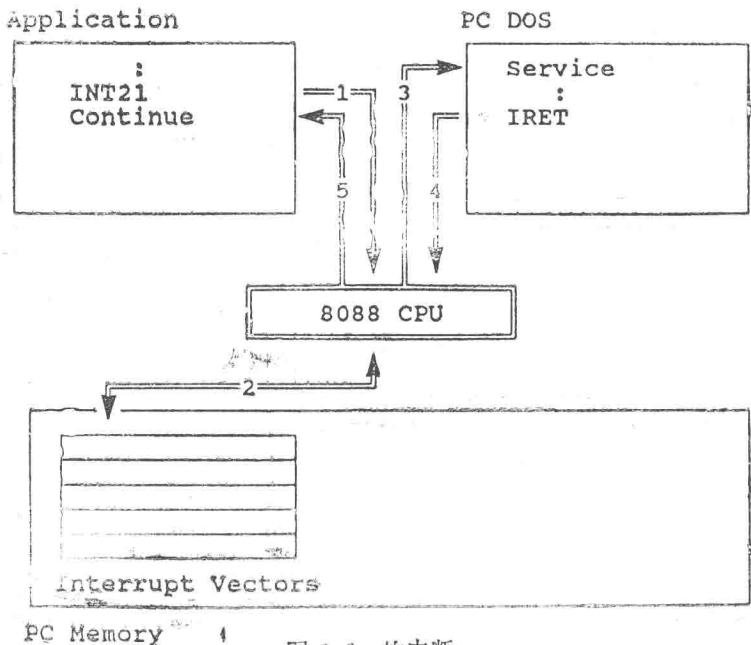


图 1-2 软中断

由CPU辅助的两个程序之间的控制权转移。

图1-2表示了一个软中断的工作过程。

1. 应用软件发出一个中断指令指向PC DOS的中断向量 (INT 21) 。
2. CPU检索驻留在内存中的中断向量表以便查找出INT 21的服务程序入口地址。
3. 通过入口地址，CPU将程序控制权移交给中断服务程序
4. PC DOS完成应用软件的一个请求后，发出IRET指令表示它已经完成了软中断的处理。
5. CPU将程序控制权返回给应用软件的下一条指令。

软中断是一种系统中的一个程序和其他一些专有程序之间的CALL/RETURN接口的方式。然而，关键问题是目标程序可以是任意的，因为CPU将程序控制权转移到中断向量中指出的程序地址。

由于PC DOS和BIOS的所有接口都通过这种机制进行，且由于存储器是未保护的，因此一个程序可以改变某个中断向量的内容以指向一个它本身的子程序。事实上，这就是大多数现有的PC DOS扩展功能赖以实现的基础。一个程序可以替换PC DOS和BIOS中断向量的内容以拦截、检查或修改系统的功能。几年来，许多很有用的PC DOS扩展软件就是用这个技术实现的。

除了有灵活性的优点外，这种扩展系统的方法有一个很大的缺陷。当你把一个以上的PC DOS扩展软件装入到系统中时，这些扩展软件之间的相互作用是不可预计的。事实上，两个或更多的程序可以拦截同一个中断向量。在扩展软件要修改服务请求时，可能导致失败。

图1-3表示了大多数可能的联接。注意，一个扩展软件可以拦截另一个扩展软件的输出。因此，用户必须解决好扩展软件的装载顺序问题。有些软件包括这样的提示：“先装载我”或“后装载我”，等等。

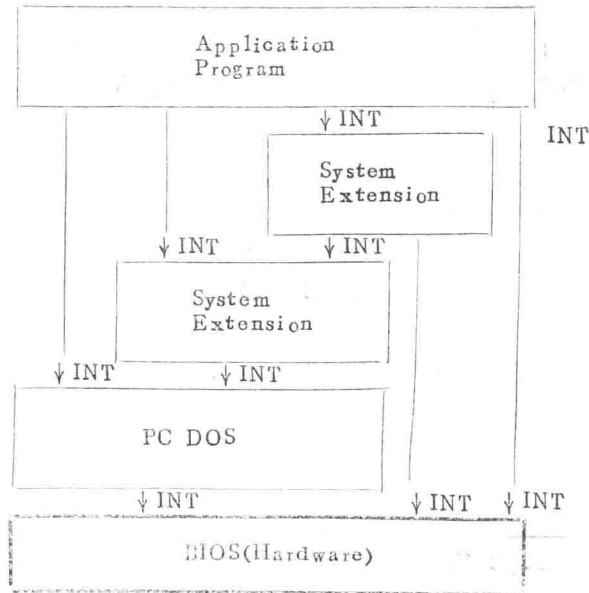


图 1-3 PC DOS的扩展

1.2 OS/2环境

PC DOS一直是PC工业演变的主要催化剂。PC DOS提供了一套基本的系统服务，成为建立新一代商业应用软件的基础。然而，由于硬件技术的进步，PC世界近几年已发生了巨大的变化。PC的作用从一种单独的机器变成一个集成化的办公室工作站。应用软件的要求已经使PC DOS超出了它本身的能力。OS/2则成为建立一种新一代PC应用环境的基础。

1.2.1 系统结构

OS/2运行于80286和80386CPU的保护模式。在这种模式下，处理器帮助操作系统完成许多功能，第二章将讨论保护模式是如何工作的。

由于是保护模式，OS/2可使它的应用软件享用一个功能明显加强了的操作环境。许多PC DOS的局限（例如640K存储器的局限）由于CPU的工作模式不同而不复存在。

1.2.2 更大的实存储器

对PC DOS应用软件来说最大的局限或许是系统的有限的存储器支持。利用80286/80386的能力，OS/2使用户可以使用高达16MB的实（物理的）存储器。与PC DOS不同，OS/2应用软件不再有640K存储器的限制。

更重要的是，这个功能使用户可以根据应用软件的需求来安装存储器。

利用纯粹的处理器支持是比利用存储器选体更为自然的突破640K局限的方法。因为应用软件不必构造覆盖结构，任何子程序可以调用应用软件中的其它子程序。

1.2.3 虚拟存储器

除了16MB是系统的物理存储器界限外，保护模式使OS/2可以实现更大的虚拟地址空

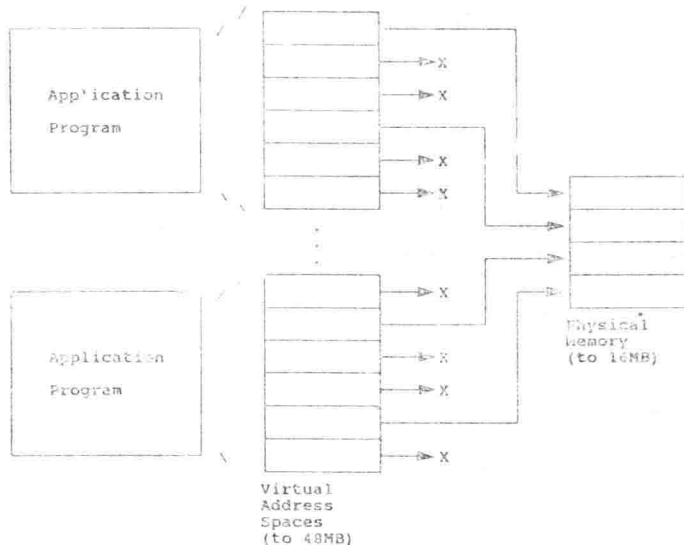


图 1-4 虚拟寻址

间。利用CPU的能力，OS/2可使它的应用软件具有独立的虚地址空间。这意味着多个程序可以共享同样的物理存储器。OS/2负责当需要时进行程序存储器和后援存储器（磁盘）之间的数据交换。图1-4表示了系统如何将多个虚地址空间映象到物理存储器中。

对应用程序来说，OS/2系统有无限的存储器。系统本身通过进行存储器和磁盘的交换隐藏了究竟安装了多大的物理存储器。这种处理对应用软件来说完全是透明的。

事实上，段交换完成了两个功能。第一，它给应用软件一个很大的“逻辑”地址空间，应用软件可以使用比机器上实际存在的存储器还要大的存储器。第二，虚地址空间为用户提供了一种更优秀的存储器结束条件。在PC DOS下，如果一个程序耗尽存储器则它本身就死亡。在OS/2下，程序可以继续运行，虽然速度会放慢。因此，OS/2使应用软件的存储器空间可以超过16MB且平滑地解决存储器越界问题。

1.2.4 存储器隔离

OS/2的存储器模式还解决了PC DOS的另一个问题：程序的隔离。所有的PC DOS软件在一个公共的存储器“池子”中运行，相互可以进行访问。这意味某个程序中的一个编程错误可能影响到另一个程序。由于OS/2运行在80286的保护模式下，因此它将每个应用软件隔离在它自己的地址空间中。

这个地址空间，是系统存储器的一个逻辑上的“子池”。一个程序只能寻址分配给它的存储器。因此，如果用户偶然（非所有权的）访问超出自己的地址空间的存储器，则OS/2可以检查出这个错误并终止该程序的执行。

考虑下面的例子。图1-5表示了三个程序（A、B和C），它们共存在存储器中。每个程序直接访问一个或多个虚拟存储器段（Segment）。尽管这三个程序使用同一个存储器，OS/2可以防止它们相互干扰。例如，假设程序A在计算时出错，试图访问程序B的一个虚拟存储段，则OS/2可以检查出这一情况并终止程序A的执行。

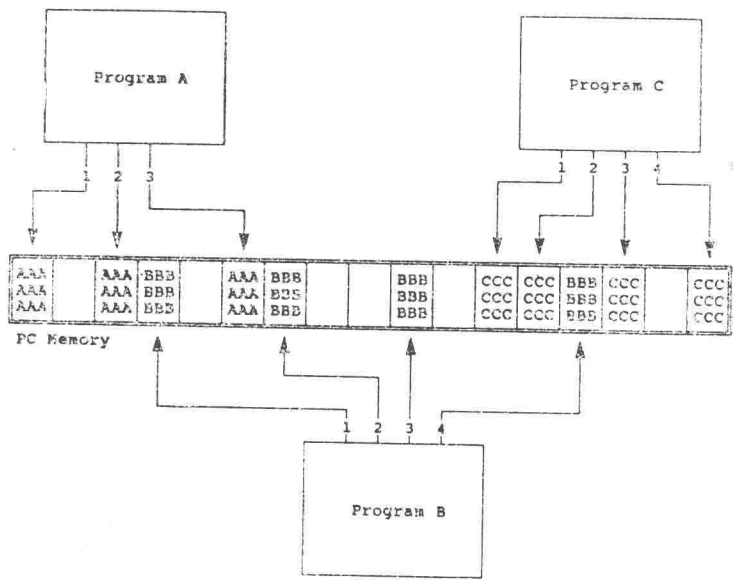


图 1-5 存储器隔离

正如程序之间彼此分开一样，系统也可与应用软件隔离。如果一个应用程序试图修改OS/2的代码或数据，则该程序可以在可能危害系统之前就被终止。因此，任何应用软件都不会使系统瘫痪。

在PC DOS下，一种典型的错误恢复方案是重新自举机器，在OS/2下则只需靠重新启动应用软件来恢复。由于系统不会因应用软件的错误而终止，因此它可以报告具体的错误信息，为用户校正应用软件中的问题提供方便。图1-6表明当应用软件出错时用户所能看到的信息。

```
Session Title: OS/2 Command Prompt

SYS1943: A program caused a protection violation.

TRAP 000D
AX=0000 BX=00C8 CX=000A DX=0000 BP=0000 SI=0008
DI=082A DS=002F ES=002F FLAGS=2246 MSW=FFFB
CS:IP=001F:0076 SS:SP=001F:01F0 ERRCD=0000
CSLIM=000C SSLIM=01FF DSLIM=0009 ESLIM=0009
CSACC=FB SSACC=F3 DSACC=F3 ESACC=F3

> End the program
```

图 1-6 异常的程序终止

1.2.5 I/O保护

OS/2的系统结构使用户可以控制应用软件与硬件打交道的方式。在PC DOS下，任何程序可以对硬件设备直接作I/O操作。由于同时只有一个程序运行因而这不是一个问题。在OS/2下，由于并发运行多个程序，因此硬件的状态必须被仔细地控制。

OS/2解决这一问题的方法是一种妥协方案。用户仍然可以在应用软件中直接进行I/O，但必须在一个受控的环境中进行。OS/2利用保护模式的功能迫使应用软件将直接I/O隔离在特定的代码段中。这些代码段必须被适当地识别，否则应用软件将失败。此外，用户可以通过系统配置时的任选参数来告诉OS/2你是否需要运行“病态”程序。在这种方式时，系统控制的I/O保护级别的权力将移交给用户进行。

1.2.6 资源管理

与PC DOS不同，OS/2承担了管理资源（磁盘、存储器、外设等等）的更积极的作用。OS/2相当于交通警察，决定这些资源什么时候是可用的和什么时候是不可用的。因为在多任务环境下资源是被共享的。

需要使用资源的应用软件必须通过OS/2的一套功能服务达到目的。在任何情况下，OS/2了解那个程序在使用着那种资源。大多数情况下，OS/2根据应用软件的要求管理资源。且即使当程序直接操作硬件时，程序也必须先通过对OS/2提出I/O请求以获得许可。因此，OS/2可解决所有的程序冲突。

OS/2系统为应用软件提供了各式各样的资源访问技术，从紧密控制（象存储器管理）