电子工业出版社
PUBLISHING HOUSE OF
ELECTRONIC INDUSTRY

MS-DOS
操作系统结构分析系列教材之一

BIO结构分析教程

郭嵩山　　编著

ISBN7-5053-1925-6/TP·460

电子工业出版社
PUBLISHING HOUSE OF
ELECTRONIC INDUSTRY

MS-DOS
操作系统结构分析系列教材之一

BIO结构分析教程

郭嵩山　　编著

ISBN7-5053-1925-6/TP·460

（京）新登字055号

内 容 简 介

本书从操作系统原理和结构的角度，采用了以模块主程序为主线，以数据结构为中心的系统软件分析方法，深入地、全面地彻底剖析了16位微型计算机上流行的MS-DOS（PC-DOS）操作系统的三大模块之一——基本输入输出模块（BIO模块）。为了便于读者学习和理解，书中从第二章起，在介绍原理和结构的过程中，对于比较复杂或重要的程序段，都配有执行流程图及这部分的全部程序详细注释清单。

本书可作为大专院校有关专业的教材和教学参考书，也可作为广大计算机工程技术人员在职培训的教材，它将是从事微型计算机系统和应用开发的工程技术人员常备的技术参考资料。

MS-DOS操作系统结构分析系列教材之一

BIO 结 构 分 析 教 程

郭嵩山 编著

责任编辑 深 民

电子工业出版社出版（北京万寿路）

电子工业出版社发行 各地新华书店经销

北京科技印刷厂印刷

★

开本：787×1092毫米 1/16 印张：25.5 字数：630千字

1992年12月第1版 1993年 8月第1次印刷

印数：1～11000册 定价：21.50元

ISBN7-5053-1925-6/TP·460

前　言

MS-DOS 是家喻户晓的 16 位个人计算机操作系统（英文缩写为 OS）。深入分析 OS 的结构，是学习和理解 OS 如何实现资源管理重要的一环，也是对 OS 进行扩充、改造必不可少的条件。例如，要研制有自主版权的操作系统和其他系统工具软件，要实现 OS 和系统软件的扩充、改造或汉化，通信及接口的改造等，对计算机病毒作用机制的解剖、病毒的防治都必须建立在对 OS 功能和结构有较为深刻的理解的基础上。所以，学习和剖析 MS-DOS 操作系统的结构，了解其实现的原理，对于从事微计算机的开发和应用人员，将会受益非浅。

本书以目前广为流行的 MS-DOS（PC-DOS）3.3 版为例，对该版本并结合其他版本的特点，详细地分析 MS-DOS 操作系统的结构和实现原理。MS-DOS（PC-DOS）3.3 版长度有 77KB，是一个比较精巧的操作系统版本，以后在此基础上发展起来的版本规模都较大（如 5.0 版就超过 100KB）。因此，读者先学习分析一种相对精小 OS，以此举一反三，有利于学习和掌握。

如何分析操作系统，笔者认为应以模块主程序为主线，以数据结构为中心的系统软件分析的方法。强调主程序为主线，可使读者快捷、省力地了解整个模块的总体结构，再逐层去剖析各个分支模块的结构。强调以数据结构为中心是因为一个系统程序的设计，在其算法确定之后，关键就是数据结构的设计。在剖析 OS 时，往往遇到的难题是对表格（线性表、链表）、缓冲区（暂存区）、静态和动态堆栈等数据结构未能弄清，对各模块所使用的数据单元的意义和取值未能了解，从而大大降低了分析和阅读系统程序清单的速度，甚至使分析工作无法进行下去。笔者积累多年在剖析和研究 OS 和系统软件中所摸索到的经验后认为，采用以模块主程序为主线，以数据结构为中心的系统软件分析方法，是一种好的分析方法。

本书对 MS-DOS（PC-DOS）面向 ROM BIOS 的下层模块，即磁盘基本输入输出模块（IBMBIO.COM 或 IO.SYS）进行了全面的深入的彻底的剖析，为了便于读者学习、分析、研究和改造，本书在第二～五章的最后一节，都列出了该部分程序的详细注释清单。由于 MS-DOS 和 PC-DOS 的相应版本在内容上几乎完全相同（除极个别指令和数据单元稍有差别外），故本书不作区分。

本书的部分内容曾以连载的方式在《电脑》杂志上发表，得到许多读者的热情支持和鼓励，在此笔者由衷地表示感谢。

本书的部分内容，曾在广州地区的一些高校的有关课程中为本科高年级学生讲授，深受欢迎。本书是根据教学实践作进一步补充、修改而成。

本书力求从有利于教与学的角度对一般人感到难度很大的操作系统内部结构和实现原理通过深入浅出的系统论述，让读者既能建立整体的概念，又能逐步深入，一层一层地剖析，为了帮助读者能结合原理读懂程序清单，我们对于程序清单的注释尽量详细，力求深入到每一条指令，而且在编排方面尽量方便读者查阅。同时，本书每章后均附有习题和思考题，以帮助读者更好地理解和掌握。

本书可作为大专院校计算机有关专业的教材和教学参考书及在职培训的教材，它将是从事微型计算机系统和应用开发的工程技术人员常备的技术参考资料。

MS-DOS 操作系统结构分析系列教材共三本，另外两本的书名是：MS-DOS 操作系统结构分析系列教材之二：《COMMAND 结构分析教程》。MS-DOS 操作系统结构分析系列教材之三：《DOS 内核结构分析教程》。

中山大学计算机科学系软件和应用专业的部分同学参加了对不同版本的 BIO 模块的剖析，做了大量的工作，他们是 85 级的陈政、陈学军、86 级的朱国庆 (3.3 版)；85 级的杨静、郑惠娟 (2.1 版)；87 级的徐朝华、符王萍、王宏书 (5.0 试用版)；此外、86 级的朱文浩、罗志良、刘南平参加了对 3.3 版的 BIO 模块的部分剖析工作。在此表示衷心的谢意。

本书是由中山大学自然科学基金资助的有关操作系统研究科研课题的成果之一，在此，向支持和赞助该项目研究的有关方面表示感谢。

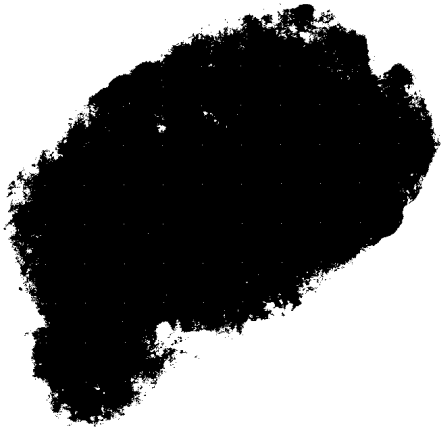
由于笔者水平所限，本书定有不少缺点和错误，恳请读者批评指正。

笔者于中山大学

1992 年 8 月

MS–DOS操作系统结构分析教材共三本：

- BIO结构分析教程
- COMMAND结构分析教程
- DOS内核结构分析教程



目　录

第0章　　绪　　论

第一节　　操作系统结构概述

早期的操作系统，由于系统规模较小，逻辑关系较简单。所以，设计者往往只注重功能设计和效率，而忽视了结构的设计。但由于操作系统设计不当，引起错误而造成惨重的损失，使人们记忆犹新。因此，人们在总结经验的基础上，认识到系统结构直接影响到系统的性能，从而越来越重视结构的设计，并逐步采用结构程序的设计方法来设计操作系统，使之成为结构清晰、易读易懂、适应性强、可靠性高、易于修改、易于证明其正确性的系统。

采用结构程序设计方法来设计操作系统。可以将操作系统看成一个整体模块，它由若干个模块按一定的结构方式组成。操作系统的结构，到目前为止，大体上可以分为三类：即模块组合结构、层次结构和管程结构。

一、模块组合结构

模块组合结构也称无序模块结构、模块接口结构，所谓模块组合结构，就是将一个大系统分成若干个相对独立的模块，这些模块可以独立编制，为使每一模块不太复杂，又可把大模块划分成更小的模块，形成"积木式"的结构方式，并把这些模块按规定的接口（如转子、调用或借助通信区、工作单元等）连接起来。

模块组合结构的特点是：

1）．模块是以功能而不是以程序或数据的特点来划分的。

2）．数据作为全程量使用。

3）．不同模块间可以不加限制地互相调用和转移，模块间信息传递方式可以随意约定。

模块组合结构的优点是结构紧密、接口简单、系统效率高，缺点是模块间独立性差，结构不清晰，不易读也不易理解，修改也不方便，往往动一处而牵动全体。此外，为了保证数据的完整性，往往采用全局封中断的方式，从而限制了系统的并发性。

早期的 UNIX 版本采用模块组合结构，UNIX 源程序划分为 44 个文件，这 44 个文件可看作成模块，其中 14 个是全局变量说明，28 个是 C 语言文件，2 个是汇编语言文件，图 0.1 示出了 UNIX 操作系统其中七个 C 语言文件间的依赖关系，其文件之间的调用是随意的，没有受一定的规律限制。

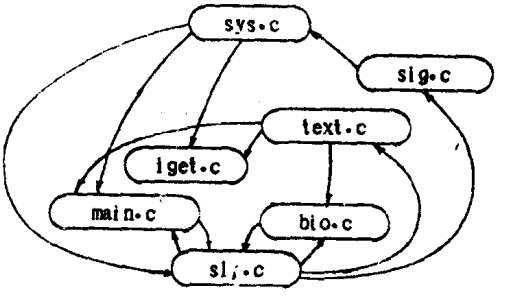


图 0.1　UNIX部分文件的调用关系

二、层次结构

所谓层次结构，就是将整个操作系统分解成若干个基本模块，并按照一定的原则，将这些模块排列成若干层，各层之间只有单向依赖关系，也即低层为高层服务，高层依赖于低层，各层之间不能构成循环。层次结构避免了模块组合结构的缺点，减少了模块间的相互依赖关系，消除了循环调用现象。

层次结构的优点是:

1) 将整体问题局部化。由于层次结构是把一个大型系统分解成若干个具有单向依赖关系的层次。因此，将对整个系统的了解分解成对各层模块的局部了解，将保证整个系统的正确性，分解成保证各层模块的正确性。

2) 层次结构的操作系统上层模块对下层模块的调用，通常设计成从统一的入口进入，也就是说，每层中各模块以统一的接口提供给上层模块调用。这样，就大大减少了接口量，从而使各层次间的调用更加清晰和规范。

3) 对于以进程作为层次中模块基本单位的层次结构（称为进程分层结构），能较好地体现了操作系统的并发特征，能动态地描述系统的执行过程。

4) 各层次间独立性强，灵活性高，易于维护、修改和移植。

5) 系统结构清晰，易于阅读和理解。

但层次结构也存在一些缺点。例如，在进程分层结构中，每个进程都要建立一个进程控制块，从而增加了系统的开销；此外，进程分层结构是由核心来统一管理控制转移、标志保留和层间的信息传递，故信息传递效率比模块组合结构低；而且，由核心统一管理，调度负荷重。

在层次结构的操作系统中，如果不仅层间是单向依赖关系，同一层间各模块也是相互独立、单向依赖和不构成循环，那么，这种层次结构称为全序结构；如果某些层次内部的模块存在着循环调用的关系，那么，这种层次结构称为半序结构。

MS-DOS 是典型的层次结构的操作系统，图 0.2 示出了 MS-DOS OS 的层次结构。

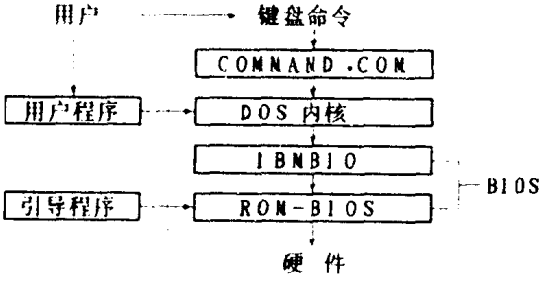


图0.2 MS-DOS的层次结构

层次结构分层的原则是以单向依赖关系为总的原则，其他的原则为:

1) 将一些直接依赖于硬件的模块（如中断处理、设备驱动等）放在最低层，这样改造后的操作系统，硬件特性消失。移植时，只需修改最低层就行。例如 MS-DOS 的常驻 IBMBIO. COM 模块等。

2) 将操作系统中指挥整个系统运转的指挥控制中心，如处理机调度、进程控制，通讯机构等模块放在最低层。

3) 将对操作系统的命令解释模块放在最高层，当要求改变操作系统的操作方式时，只要改变最高层就成，例如 MS-DOS 的 COMMAND. COM 模块等。

4) 将接受控制的模块放在较高层，并将非资源分配而又与硬件特性关系不大的文件管理模块放在中间层，例如 MS-DOS 的 IBMDOS. COM 模块就安排在中间层。

5) 对被调用的模块尽量放在靠低层，对于专用模块（不被其他模块所调用）放较高层，以符合高层向低层请求服务的单向依赖关系的原则。

6) 将同类或相似功能的模块尽量放在同一层，例如，存贮管理、文件管理、设备管理等模块，一般都分别放在同一层，尤其不要将有相近功能的模块放在相隔的两层中。

层次结构的操作系统设计方法通常有自底向上法（bottom-up）和自顶向下法（top-down）两种，前者是从硬件或经软件扩充的第 1 级虚拟机底部开始逐层扩充功能，直至到达目标系统，后者则恰好相反。

三、管程结构

1、管程和类程概念的引入

1) 管程 (monitor)

由于各进程在共享临界资源时必须互斥，每次只允许一个进程进入临界区，为此，各个使用临界区的进程必须使用同步操作。此外，为实现异步环境下进程间的通讯，也需要同步操作。这样，使大量同步操作分散于各进程中，而同一进程，需要使用多个临界资源时也需要若干个同步操作。如此多的同步操作分散在各个进程中，往往会因为使用不当而发生死锁。为解决共享资源同步操作分散在各个进程而引起系统可靠性方面的问题，而产生了将共享资源全部同步操作集中在一个程序单位的设想。在用数据结构抽象表示共享资源时，资源管理程序就可用在该数据结构上进行操作的一组过程来表示，从而引入管程（资源管理程序）的概念。所谓管程，是指共享资源的数据结构以及在其上的能为并发进程所执行的一组操作。

一个管程由以下四大部分组成:

① 管程名 Monitor;

② 局限于管程的数据说明;

③ 在该数据结构上进行操作的一组过程;

④ 对局部数据赋予初值的语句（初启语句)。

管程与进程的区别在于:

① 管程定义的是公用数据结构，进程定义的是私用数据结构;

② 管程定义的是在数据结构上的同步操作和初始操作，并将系统的同步操作相对集中起来，而进程定义的是顺序操作;

③ 管程是为解决进程共享资源的互斥而设置的，进程是为实现系统并发性而设置的;

④ 管程被进程所调用，管程和调用它的进程不能并行操作，而进程间可以并行操作。

⑤ 进程有生命期，由创建到撤消，管程是控制系统所固有的，不需由进程创建或撤消，只供进程调用。

2) 类程 (class)

所谓类程是指专用资源的数据结构以及在其上规定的全部操作。由于类程是在专用资源上进行操作的一组过程，所以它不存在同步操作。

一个类程由以下四大部分组成:

① 类程名 class;

② 局限于类程的数据说明;

③ 在该数据结构上进行操作的一组过程;

④ 初启语句。

管程与类程的主要区别在于管程是管理共享资源，将竞争共享资源的并发进程通过同步操作处理成顺序执行。类程是管理专用资源，类程被进程所调用，被看作进程的延伸，不同的进程调用各自的类程。

2、管程结构操作系统

从系统功能和实现相结合的观点出发，从系统中提炼出管程、类程、进程等几种基本成分，将系统分解成由这些基本成分组成的模块，并将这些模块按一定的原则编入各层，核心在最内层，看作是管理 CPU 的专用管程，这种结构称为管程结构（也称为层次管程结构）。这种结构具有以下特点:

1) 从数据结构的角度来看，它是将数据结构及其上操作集中起来的一种抽象的数据类型;

2) 从资源管理的角度来看，它将系统分成若干模块，用数据表示抽象的系统资源，并根据共享资源和专用资源在管理上的差别来定义模块的类型和结构，从而引出了管程和类程的概念。它使系统的同步操作相对集中，从而增加了模块的相对独立性。

第二节　分析操作系统的方法

一、分析前的准备

要分析 MS-DOS 操作系统，在分析前需要作如下的准备工作。

1、熟悉 MS-DOS 的原理和结构

要分析一个操作系统，首先要熟悉它的基本原理、总体结构，各模块的内存分配以及各种基本命令的使用方法，以便于阅读和分析操作系统。同时要熟悉基本工具，例如，MS-DOS 的 DEBUG.COM 的使用方法。

2、阅读操作系统各模块的程序清单

如果我们手头有一些现成的系统分析软件，例如: SOURCER、ASMGET 等，当然是最好不过的了。但如果没有，就只能用系统配备的实用程序 DEBUG.COM 了。

使用 DEBUG.COM 来阅读 MS-DOS 程序清单，首先要先将作为隐含、只读和系统文件的 IBMBIO.COM（或 IO.SYS）及 IBMDOS.COM（或 MSDOS.SYS）转换成可读写的文件，其操作是将其目录区中的文件属性进行修改。下列以 PC／XT 为例，首先将 MS-DOS（3.3 版）系统盘插入 A 驱动器中。然后执行下列操作（有下横线者要从键盘键入）。A＞DEBUG＜CR＞

-lCS：100 0 5 7＜CR＞　　；读目录

-dCS：100<CR>　　;显示目录
-eCS：10B<CR>　　;修改属性
0FC3：10B 27.20<CR>
-eCS：12B<CR>
0FC3：12B 27.20<CR>
-dCS：100<CR>　　;显示目录
-WCS：100 0 5 7<CR>　　;写回目录区
-Q<CR>
A>

在修改了两个系统文件属性后就可用 DEBUG 装入 IBMBIO.COM 或 IBMDOS.COM 文件，用 U 命令反汇编读出指令码，用 D 命令读数据，

其使用格式是:

U<始地址><终止地址>

D<始地址><终止地址>

其中始地址包括段址：段内偏移。

3、分清指令单元和数据单元

可以根据指令的特征，用人工跟踪的办法，大体分清指令单元或数据（工作）单元。

1）数据单元的区分

凡是使用牵涉到寄存器和存储器之间数据传送指令，所用到的内存，一定是数据单元或工作单元，例如使用 MOV 指令传送到的内存。此外，用 D 命令列出的有明确意义的字串，则肯定是数据单元。

2）指令单元的区分

凡转移指令或调用子程序指令所指示的转移地址及入口、出口，都是指令单元。根据上述方法，在屏幕上扫描整个待分析程序，记下指令单元和数据单元的始末址。

4、将待分析程序转换成磁盘文件

为了更好地阅读和注释，可以用管道和改向操作命令将待分析的程序转换成磁盘文件，这样，以后分析就方便多了。其操作示例如下:

1）建立 DEBUG 子命令文件

首先在控制台上建立所用来列出指令或数据的 DEBUG 子命令文件，取名为 DEBUG1。

A>COPY CON DEBUG1
UXXXX：YYYY ZZZZ<CR>　　;XXXX：YYYY 为指令单元始址，ZZZZ 为指令单元末址
DAAAA：BBBB CCCC<CR>　　; AAAA：BBBB 为数据单元始址 CCCC 为数据单元末址
;根据需要可以用多个 U、D 命令
Q<CR>
^Z<CR>

A>

2) 建立待分析程序的反汇编磁盘文件

插入有操作系统 IBMDOS.COM、IBMBIO.COM 或 COMMAND.COM 文件的磁盘到 A 驱动器，执行下述操作:

A>TYPE DEBUG1 | DEBUG IBMBIO.COM〉 B: IBMBIO.TXT

在操作时特别要注意的是反汇编后的文本文件比原来的大许多，为将来编辑及注释方便可将系统文件拆成几个反汇编后的磁盘文件。

示例中获得的清单是直接反汇编系统文件而得到，至于需要获得 OS 各模块在内存的实际工作位置中的程序清单，也即象本系列书中所列出的程序注释清单，则需要了解 OS 各模块装入内存的实际段址，详细将在本书后面各章中进行介绍。值得一提的是，除了数据单元中个别数据在运行时有所改变外，其指令代码是完全一样的（除根据系统不同配置而作删除的以外），只不过是定位的段址不同罢了。

5、打印出较为准确的程序清单

在第 3～4 项工作的基础上，便可以用行印机印出较为正确的程序清单。当然，也不排除个别数据或工作单元仍误为指令单元，但"漏网"者毕竟是少数，不会影响分析工作的进行，并可在分析过程中不断予以纠正。在获得比较正确的程序清单后可以转入分析工作。

二、分析操作系统的方法

分析操作系统建议采用笔者所倡导的以模块主程序为主线，以数据结构为中心的系统软件分析的方法。

1) 根据操作系统的功能，采用模块分解的原则，将其划分成各自相对独立子模块来分析。

在程序设计中,将一个具有确定功能的程序段称作模块.模块有入口和出口.各模块间的联系,只要根据模块要求,了解模块功能.给出入口参数并了解其返回参数（出口信息），就可以调用此模块。在分析 OS 时，要根据程序设计时模块化设计的原则，将 OS 划分为多个模块，在分析深入的过程中，逐步将每个子模块再划分成更小的相对独立的小模块。也就是说，把分析一个大系统，分解成分析若干个小系统，只要将各个小系统（小模块）与大系统（大模块）及其他小模块之间的联系搞清楚,就可以将一个复杂大问题拆成一个个小问题去解决.采用这种分解模块的方式,就不会在一个大的系统程序之前束手无策,无从入手。

2) 在分析每个模块中,要从主程序入手,逐步深入,逐步精细化地进行分析。

我们知道,在程序设计时,要运用《程序设计方法学》的原理。采用自顶向下逐步求精的方法。而在分析操作系统时，应从主程序入手，尽量抛开支节，逐步深入，逐步补充完整。

3) 注意各模块的数据结构和控制结构。在 MS-DOS 操作系统中，大量使用的数据结构是表、缓冲区和栈。缓冲区实际上是一种特殊的队列，只对它进行读/写操作，与队列不同的是，缓冲区只需要一个指针指示对其读/写信息；而对静态栈，尤要熟记栈顶（底）指针，同时要注意动态栈的结构。在程序的控制结构中，大量用到的是二叉树结构，在分析过程中尤其注意产生分枝的条件转移指令、条件转子指令和调用子程序指令。

同时，要注意多叉树枝叶的走向，这主要是使用CALL　DX等指令时出现的。

4）注意数据单元和工作单元的初值或意义，最好预先列好表，并在分析过程中不断充实和修正。

5）注意各程序段的功能，逐步完善各模块中各程序段的内存分配表，在分析过程中，注意上述第3～5点，对于加快分析进度、提高分析质量将有很大的帮助。

6）对于较大规模的系统分析，建议先研制适合机型的分析工具软件，能够打印出自动分清数据单元和工作单元的反汇编清单（一般操作系统所提供的DEBUG程序反汇编时均不能做到这一点）并能初步赋标号，以加快分析的速度。

三、操作系统分析的要求

如何才算完成对一个操作系统或其实用程序等系统软件的分析工作呢？笔者认为应完成如下工作：

1）将被分析系统的程序清单全面加标号和注释，也即完成被分析系统详细的注释清单。

2）在分析过程中，绘制出主程序及主要子程序的执行流程图。

3）完成被分析系统各个功能程序段及子程序、工作单元和数据单元的内存分配表。

4）将注释内容输入到反汇编文本文件中，以完成注释文件的整理工作。

5）编写分析报告。

第三节　MS-DOS概述

一、MS-DOS的发展历史

1980年，Microsoft公司将86-DOS经过较大的改进后，取名为MS-DOS，并将它提供给IBM公司作为IBM-PC机的基本操作系统，取名为PC-DOS。这就是DOS 1.0版。MS-DOS（PC-DOS）1.0版基本上沿用了8位微型机CP/M OS的许多功能，如在文件管理，文件控制块（FCB）结构，程序段前缀（PSP）和可执行文件（COM）等结构上继承了CP/M OS的风格。1.0版并没有充分发挥8088 CPU的特点，实际上它只不过是CP/M OS的8088版本。到了1983年，为支持带硬盘的PC/XT机，IBM公司将MS-DOS版本1经过较大的改造，吸取了UNIX OS的许多优点，形成了MS-DOS 2.0版本。1984年，为适应支持1.2MB软盘的PC/AT机的需要，MS-DOS版本2被升级为版本3.0。以后不久，为适应网络的需要，又推出了3.1版本。到1986年，为支持3.5英寸软盘的要求，又推出了3.2版本，但是无论3.0，3.1还是3.2版本，都未能充分发挥80286 CPU的功能。1986年11月，推出了支持多任务（多进程）的MS-DOS 4.0版，该版本在3.2版的基础上发展起来的，能运行前后台程序，并具有支持网络的功能。到了1987年，为兼容IBM PC系列机和PS/2个人计算机系统，IBM公司又推出3.3版本，直到目前，3.3版仍是较受用户欢迎的流行的版本。1988年7月，IBM公司为了改善用户介面，推出带有DOS-SHELL及下拉式菜单的PC-DOS　4.00，它允许文件系统所用盘空间超出32MB，1989年又推出改进后的PC-DOS　4.01版，但由于PC-DOS　4.00内核扩展太大，使许多应用软件由于内存不足而不能运行，致使相当数量的网络软件无用武之地，其SHELL功能也无特色，有的用户批评它甚至比不上

PCTOOLS。1990～1991 年 Microsoft 推出了先是试用版后是标准版的 MS-DOS 5.0，它支持 2.88 兆字节的软盘驱动器，提供 DOS 命令及 SHELL 的联机帮助（这是 3.3 版受批评最多之处），能够在高址内存（HMA）运行，具有改进的图形接口，最大可建立 2000 兆字节的磁盘分区，其配备的编辑器可进行全屏编辑，由于 MS-DOS 5.0 的内核小巧精细（与 3.3 版相差不大），因而 5.0 版的推出受到用户广泛的欢迎，但是 MS-DOS 5.0 充其量只是一个以 8086／8088 为基础，较好地利用 80286 功能的 OS，仍然未能充分发挥 80386／80486 CPU 的特点。为便于读者直观了解。我们用图 0.3 示出了 MS-DOS 的发展过程。

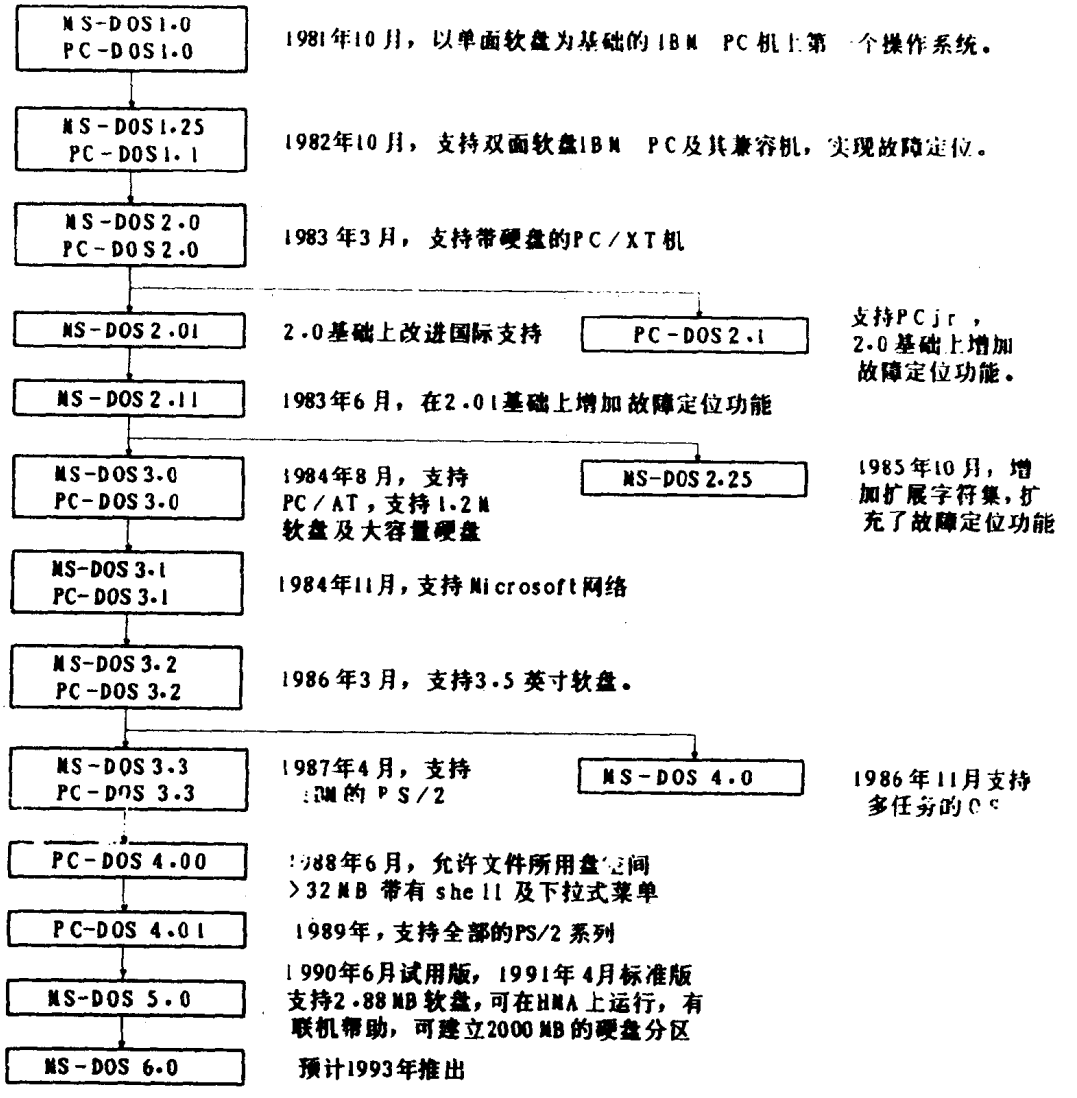


图 0.3 MS-DOS（PC-DOS）的发展图

从 MS-DOS 整个发展过程来看，可以归纳出如下特点：

1、MS-DOS（PC-DOS）的版本更新很快，短短十年，就更新了 15 次版本，这正反映了 80 年代以来软件技术的蓬勃发展的特点，在硬件迅速发展，竞争十分激烈的情形下，不及时更新，势必会被淘汰。

2、MS-DOS 的版本更新，多数是适应磁盘存储器的发展而发展，从支持单面、双面软盘到支持硬盘，从低容量软盘到高密度、大容量软盘，从支持 5.25 英寸软盘到支持 3.5 英寸软盘。每一次外存储器发展，都推动了 OS 版本的更新。

3、包括 OS 在内的系统软件的发展都跟不上由于微电子技术迅猛发展而引起的硬件飞跃的进步，每次新机型出现，为之配套的 OS 版本总是未能充分发挥其新发展硬件的优点，这是采用先研制硬件再为其配套软件的发展策略的弊病。

二、MS-DOS 的组成

MS-DOS 操作系统是由三个独立的而相互间又有联系的程序模块以层次结构的方式组成。这三层模块分别是：

1、DOS-BIOS 模块，其文件名为 IBMBIO.COM（或 IO.SYS）。

2、DOS-Kernel 模块，其文件名为 IBMDOS.COM（或 MDOS.SYS）。

3、DOS-Shell 模块，其文件名为 COMMAND.COM。

其中，DOS-Shell 模块是面向用户的，它对用户输入的内部命令或外部命令进行读取、识别、处理和执行。在其运行过程中要调用到 DOS-Kernel 模块内相应的功能。Kernel 模块是 DOS 的核心，在实际完成指定的功能时，DOS-Kernel 模块通过调用 DOS-BIOS 模块的相应子程序驱动固化在 ROM-BIOS 中的管理硬件的中断例程。这样，MS-DOS 操作系统的层次结构确保 DOS 核心在运行时，与硬件环境在逻辑上是完全隔离的，这有利于用户编制的应用程序，能不依赖于具体的硬件设备，而能在配备 MS-DOS 的各类微机上运行。这就是说，隐含文件 IBMBIO.COM 是 DOS-Kernel 与 ROM-BIOS 的接口模块，提供了对于 ROM-BIOS 设备驱动子程序的低级接口。MS-DOS 三个基本模块的长度如表 0.1 所示。

表 0.1　MS（PC）-DOS 基本模块长度

基本文件	2.0	2.1	3.0	3.1	3.2	3.3	5.0 试用版	5.0 标准版
IBMBIO.COM	4608	4736	8964	9561	16396	22100	31833	33430
IBMDOS.COM	17152	17024	27920	27760	28477	30159	36348	37394
COMMAND.COM	17664	17792	22042	23210	23791	25307	41035	47845
占用磁盘空间	39424	39552	58926	60531	68664	77666	109216	118669
系统程序占用区	约 25K	约 25K	约 37K	约 37K	约 46K	约 54K	约 84K	约 92K
暂态区	约 13K	约 13K	约 17K	约 18K	约 18.5K	约 19K	约 32K	约 38K
所占内存空间	约 38K	约 38K	约 54K	约 55K	约 65K	约 73K	约 116K	约 120K