

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二

COMMAND 结构分析教程

郭嵩山

编著

ISBN7 - 5053 - 3463 - 8/TP · 1370

TP316
298

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二

COMMAND 结构分析教程

郭嵩山

编著

ISBN7 - 5053 - 3463 - 8/TP · 1370

内 容 简 介

本书从操作系统原理和结构的角度,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,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——用户命令处理模块(COMMAND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高层次在职培训的教材,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二

COMMAND 结构分析教程

郭嵩山 编著

责任编辑 王惠民

电子工业出版社出版(北京万寿路)

电子工业出版社总发行 各地新华书店经销

电子工业出版社广州科技公司排版

北京市顺新印刷厂印刷

★

开本:787×1092毫米 1/16 印张:16.5 字数:350千字

1996年8月第1版

1996年8月第1次印刷

印数:2000册

定价:25.00元

ISBN7-5053-3463-8/TP·1370

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,要研制有自主知识产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、病毒的防治以及对系统数据的保护和维护等方面都必须建立在对 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的开发人员,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)面向用户的上层模块,即用户命令处理模块(COMMAND.COM)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 and 掌握。

本书可作为大专院校计算机有关专业本科高年级学生及研究生的教材和教学参考书以及广大科技人员高层次的在职培训教材,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。

在教学上建议根据专业的性质,选择 DOS 三大模块中的一个模块进行精讲,并扼要讲授其他两个模块的主要结构和特点,其他内容可由学生去阅读。笔者从多年的教学实践中深切体会到,让学生完整地读通一个实用的操作系统或其主要模块,对提高他们的系统分析和设计能力,确有很大的帮助。

MS-DOS 操作系统结构分析系列教材共三本,其中 MS-DOS 操作系统结构分析系列教材之一:《BIO 结构分析教程》已由电子工业出版社已经出版发行,另一本 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》与本书同时由电子工业出版社出版发行。

中山大学计算机系软件专业的部分同学参加了对不同版本 COMMAND 模块的剖析,做了大量的工作,他们是 85 级的周洪宇、李杨桃、87 级的黄晓芬、罗立红、符靖(3.3 版);87 级的黄晓芬、罗立红、杨阳春、刘恒(5.0 试用版),88 级的何燕云、林萍、林燕、范志群(5.0 标准版),86 级的陈卫东、87 级的杨阳春也参加了对 3.3 版的 COMMAND 模块的部分剖析工作,在此表示衷心的感谢。

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

1995 年 10 月

目 录

第一章 COMMAND 模块总体概述	(1)
第一节 COMMAND 模块的总体结构	(1)
第二节 COMMAND 模块的数据结构	(3)
习题和思考题	(12)
第二章 COMMAND 模块初始化程序	(13)
第一节 初始化程序的功能和内存组织	(13)
第二节 初始化程序功能实现原理	(16)
第三节 程序注释清单	(21)
习题和思考题	(36)
第三章 COMMAND 模块常驻程序	(37)
第一节 常驻程序的功能和内存组织	(37)
第二节 DOS 专用的中断和中断退出的实现原理	(42)
第三节 装入暂驻程序的实现原理	(48)
第四节 程序注释清单	(48)
习题和思考题	(66)
第四章 COMMAND 模块暂驻主程序	(67)
第一节 暂驻主程序结构概述	(67)
第二节 暂驻程序的数据结构和内存组织	(69)
第三节 命令的读取与识别	(84)
第四节 命令的处理	(90)
第五节 管道命令和输入输出改向命令的处理	(95)
第六节 程序注释清单	(98)
习题和思考题	(116)
第五章 DOS 内部命令实现原理	(117)
第一节 概述	(117)
第二节 目录操作	(118)
第三节 文件操作	(123)
第四节 开关操作和环境设置	(131)
第五节 批命令操作	(137)
第六节 I/O 操作、日期时间操作和其他操作	(144)
第七节 程序注释清单	(148)
习题和思考题	(258)

第一章 COMMAND 模块总体概述

第一节 COMMAND 模块的总体结构

一、COMMAND 模块的组成

MS-DOS 的 COMMAND 模块由初始化程序、常驻程序和暂驻程序这三个相对独立的部分组成。MS-DOS 操作系统启动的过程是：开机时，由引导程序将 IBMBIO.COM 文件读入内存 0070: 0000H 处，并执行 IBMBIO.COM 的初始化程序。该程序运行时，确定内存容量，并将 IBMDOS.COM 文件读入 IBMBIO 区域之后的内存，并调用 IBMDOS.COM 的初始化程序，将 DOS 内核定位于内存之中，IBMBIO 初始化程序还执行解释 CONFIG.SYS 文件，建立操作系统运行环境，加载可安装的设备驱动程序等任务。接着，将 COMMAND.COM 文件装入内存 CS: 0100H 处，然后将控制权交给 COMMAND.COM 的初始化程序，执行 COMMAND.COM 初始化程序，完成命令处理前的初始化工作，并将 COMMAND.COM 的暂驻部分装入内存 (RAM) 高地址处 (紧靠位于高址处的 ROM 之前)。检查 AUTOEXEC.BAT 文件存在否，如有，执行之。否则转入系统命令状态，等待用户发出的操作命令。有关 MS-DOS 启动的详细过程请参阅笔者所著的《BIO 结构分析教程》一书第一章。

MS-DOS 的三个基本模块中的初始化程序都是在执行各自任务完成后自行撤消，不常驻在内存中。

二、内存分配

MS-DOS 装入内存后，其在内存的分配如表 1.1 所示。

表 1.1 MS-DOS 的内存分配

起始地址	内 容
0000: 0000H	中断向量表
0040: 0000H	ROM 通讯区
0050: 0000H	DOS 通讯区
0070: 0000H	IBMBIOS.COM 常驻部分 (BIO 标准设备驱动程序)
	IBMDOS.COM 常驻部分 (DOS 内核)
	操作系统运行环境及可安装的设备驱动程序
× × × ×: 0100H	COMMAND.COM 常驻程序
CS: 0E30H (3.3 版) CS: 1560H (5.0 版)	COMMAND.COM 初始化程序 (初始化完成后，释放该空间给用户区)
× × × ×: 0100H	用户区
× × × ×: 0100H	COMMAND.COM 暂驻程序 (此段空间在运行用户程序时可被复盖)
A000: 0000H	存放在 ROM 中硬盘驱动程序及显示 ROM 等
F000: 0000H	ROM BIOS

上表中 COMMAND 模块常驻程序段址可通过查中断向量表 2EH 类中断程序入口来确定，暂驻程序段址可通过查常驻程序中存放该值的工作单元来确定，表 1.2 列出了 3.3 版本的 COMMAND 各部分的段址。

表 1.2 COMMAND 各部分段址的确定

常驻段址	[0000;00BA]
初始化段址	CS
暂驻段址	(常驻段址;0BDF)

MS-DOS 内存管理的一个重要特征是通过分段的方法, 实现程序的重定位, 所谓重定位, 就是同一程序代码 (包括所处理的数据) 能装入到内存“任意”区域, 而不必修改代码本身。暂驻程序重新装入到内存高地址处, 就是根据这个特性, 通过改变段址来实现。

打印 COMMAND 模块清单, 可用 DEBUG 命令将 COMMAND.COM 文件装入到内存用户区, 在 DEBUG 状态下, 用 U 命令打印程序代码清单, 用 D 命令打印数据 (工作) 单元清单。其中, 初始化程序直接从用户区打印, 常驻程序既可从表 1.2 所列的常驻段址 (即实际工作区域) 开始打印, 也可以从用户区 (段址为 CS) 开始打印, 暂驻程序从它实际工作的内存高地址处打印, 其内存分配如表 1.3 所示。

表 1.3 3.3 版 COMMAND 的内存分配

项 目		段内偏移
常驻部分	程序代码区	100—7BBH
	数据(工作)区	00—FFH, 7BC—E2FH
初始化部分	程序代码区	0E30—145CH
	数据(工作)区	145D—175FH
暂驻部分	程序代码区	100—3728H
	数据区	3729—3F43H
	工作单元区	3F44—4E6BH

在表 1.3 中, CS 为用户代码段址, 其值由装入时决定, 暂驻程序段址可通过查常驻程序中存放该值的工作单元来确定 (见表 1.2)。对于不同的硬件配置, 同一程序装入的段址是不同的。

三、COMMAND 模块的功能

COMMAND 模块由初始化程序、常驻程序和暂驻程序三部分组成, 这三部分既相对独立又互有联系。

初始化程序的功能是完成命令处理前的初始化工作, 如设置一些指针, 保留一些必要的数, 修改中断处理程序入口, 将暂驻程序装入内存并对自动批文件进行处理, 初始化程序在完成上述功能后, 自行撤消, 释放本身所占的内存空间, 将控制转移到常驻程序。

常驻程序包括暂驻程序执行时要调用到的子程序和数据, 它不象初始化程序和暂驻程序那样有一个执行主体任务的主程序, 而是由一些各自相对独立的子程序所组成, 其中主要的有, 各类中断处理子程序, 装入暂驻部分及与之相关的子程序。常驻程序在操作系统工作过程中一直驻留在内存。初始化程序在完成自身任务后, 将控制转移到常驻程序中的

程序终止处理（即 22 类中断处理）子程序，该子程序在进行一系列状态处理后，再把控制转移到暂驻主程序的入口。

暂驻程序是 COMMAND 模块的核心，承担对用户命令进行处理的任务，它包括命令读取和识别，内部命令处理和执行，外部命令（包括批命令和外部可执行命令）处理和执行等功能。顾名思义，暂驻程序仅在命令处理过程中暂时驻留在内存，在 OS 执行其他用户作业时，可以将暂驻程序复盖，以腾出更大的内存空间供用户作业用，在用户作业退出时，才由常驻程序中装入暂驻部分子程序将暂驻程序重新装入到内存（RAM）的高地址处。

第二节 COMMAND 模块的数据结构

从资源管理的观点来看，计算机操作系统是管理计算机硬、软件资源的系统程序。操作系统要对资源进行管理，需要用数据来抽象表示系统中的各种硬、软件资源。例如，用变量表示某类外设，用数组表示缓冲区，用记录表示表格等等，系统程序作用对象就是数据。著名的瑞士计算机科学家沃斯（N.Wirth）教授提出的“算法+数据结构=程序”的著名论点，简洁地概括了程序的组成。一个系统程序的设计，在算法确定后，关键就是数据结构的设计，描述一种系统资源，往往可以选取多种数据结构，而数据结构的选取在很大程度上决定了程序设计的难易、效率、易读性和可适应性。因此，掌握一个系统程序的数据结构以及在其上的操作，就能掌握这个系统程序的核心和关键。这也是本书以较多篇幅描述数据结构的原因。

3.3 版的 COMMAND 模块装入内存后，其数据块有 4 块，它们是：程序段前缀（0000~00FFH）、常驻程序数据区（7BC~E2FH）、初始化程序数据区（145D~175FH）和暂驻程序数据区（3729~4E6BH）。

以上括号内标出的是段内偏移地址。在这里，特别要注意的是，暂驻程序与常驻程序实际工作时的段址是不同的。

COMMAND 模块用到的数据结构包括表格、栈和缓冲区，对 MS-DOS 不同版本，其 COMMAND 模块所用到的数据结构基本相同，只不过是它们在程序中的偏移地址不同罢了，下面分别介绍。

一、表格

表格是 MS-DOS OS 重要的数据结构，COMMAND 模块大量用到的表格是线性表。所谓线性表是指表中每个元素，除第一个外，都仅有一个直接前趋（Predecessor），同时除最后一个外，都仅有一个直接后继（Successor），后述的栈和缓冲区，实际也是一种特殊的线性表。COMMAND 模块中用到的公共表格包括程序段前缀（PSP）、文件控制块（FCB）、文件目录项（FDT）、文件分配表（FAT）等。

1. 程序段前缀表（PSP）

程序段前缀（PSP）在内存 CS: 0000~00FFH，它保存了系统模块及用户程序用到的一些中断向量、数据以及工作区。用 DEBUG.COM 程序将 COMMAND.COM 装入

内存程序段前缀的例子请参看第二章程序注释清单最开头的部分。

程序段前缀 (PSP) 表结构如下表 1.4 所示, 其中带 * 的, 是目前一般文献尚少作介绍的。

表 1.4 程序段前缀(PSP)

0	1	2	3	4	5	6	7	8	9	0A	0B	0C	0D	0E	0F
INT20 指令		可用内存最高节数		保留	DOS 长调用					INT22H 入口地址偏移值 (IP)		INT22H 段址 (CS)		INT23H 入口地址偏移值 (IP)	
10	11	12	13		段内可用节数					1A	1B	1C	1D	1E	1F
INT23H 段址 (CS)		INT24H 入口地址偏移 (IP)		INT24H 段址 (CS)		父程序段址*		系统打开文件表*							
20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
系统打开文件表											环境块段址		用户程序堆栈指针 (SP)		
30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
用户程序堆栈指针 (SS)		文件表* 长度		文件表* 偏移		文件表* 段址		保 留							
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
保 留															
50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
INT 21 指令		R E T 指令		保留		扩展的 FCB1*					未打开的 FCB1				
60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
未打开的 FCB1											未打开的 FCB2				
70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
未打开的 FCB2															
80	81~FFH														
命令行长度	命令行 / 磁盘读写缓冲区														

下面, 对 PSP 表中一些重要的域加以说明:

1) INT20H 指令

它在 PSP 表 00~01H 处, 它实际是 DOS 提供的程序终止处理的中断例程, 执行时自动恢复 INT22H, 23H 和 24H 原向量地址, 并释放被加载程序所占有的内存资源, 将控制返回 COMMAND 暂驻主程序入口。

2) 可用内存最高节数

MS-DOS 是以节 (paragraph) 为单位进行内存管理的, 1 节 = 16Bytes 即 16 进制的 10HB。它在 PSP 表的 0002HWord, 表示当前可用内存的最高端段址。在本例中, (0002H) = A000H 即 640KB。此域对 COM 和 EXE 文件有不同的意义。

如果是 COM 文件, 在它装入时分配掉所有可能的内存, 此时本域实际上包含的是用

户内存顶部的地址（除非该程序有意释放过内存）。从而在此地址之上已不能再获得内存。

EXE 文件不一定分配完所有的内存，故可作通过执行 MS-DOS 48H 号功能调用获得内存顶部地址上面的额外内存，初始分配给 EXE 文件的内存量取决于文件头最大节 (maximum paragraphs) 域中的数值。

3) DOS 长调用操作码

该域偏移量为 05~09H，其中 05H 为 CALL 指令码 (9AH)，06~09H 为长调用地址值，这是 DOS 低版本为了兼容 CP/M 的调用方式，使用目前版本应避免使用，而将位移 06~07H 作为段内可用字节数，本例中其值为 FFF0H，即 64KB 扣除 COM 文件栈区值。

4) DOS 专用中断入口

该域偏移量 0A~0DH、0E~11H、12~15H 分别存放 INT22H、INT23H、INT24H 的原先内容。

5) 父进程段址*

它的偏移量为 16H，此域包含创建此进程的父进程的段址。

6) DOS 文件表*

此域的偏移量为 18H，共计有 14H=20 个字节。

DOS 提供“打开”文件的操作，其返回的文件句柄为 0 到 19 之间的一个数，并在 PSP 偏移量 18H 处的 20 个字节文件表上作一标引。DOS 文件表中每个单字节的登记项如果包含的值是 FFH，说明此表登记项未使用。如果数值为 00 至 FEH，则为 DOS 内部使用的表示使用这个实际文件或设备的数码。此数码被称为“DOS 内部数码”以区别文件句柄，一个文件句柄传给 DOS 后，DOS 并不使用该句柄去标识实际的文件，而使用的是从文件表中得到的 DOS 内部码，内部码有以下数值（表 1.5）。

表 1.5 DOS 文件表内码值及其意义

内部码	对应的文件/设备
00	辅助设备(AUX)
01	控制台设备(CON)
02	打印机(PRN)
03 及以上	由程序打开的文件/设备

实际上，这个 DOS 内部码是相应的句柄参数块号。这个参数块号指明控制该设备/文件的句柄参数块。

在 DOS 中，00、01、02 号句柄参数块分别对应标准辅助设备、控制台设备、标准打印机设备的句柄参数块。而 03 号以上的则对应用户程序打开的其他文件或设备的句柄参数块。为 FFH 时则该句柄参数块未使用。

一个程序装入时，DOS 文件表具有下面的初始值（表 1.6）

表 1.6 DOS 文件表

偏移量	句柄名称	值	缺省设备
18H+00	标准输入	01	键盘
+01	标准输出	01	显示器
+02	标准错误	01	显示器
+03	标准辅助设备	00	COM1:
+04	标准打印机设备	02	LPT1:
+05 及以上	自由句柄	FF(未打开)	

DOS 文件表的前 5 个单元对应标准的预先打开的文件句柄。若没有必要则不要修改它们。标准输入、输出和错误设备均指控制台设备。

此方法的优点是 DOS 可由简单地改变文件表中的对应登记项来重定向一个文件句柄。例如，偏移量（DOS 文件表的偏移量）01 处为标准输出文件句柄。通常包含用于控制台（01）的内部数码，如果操作系统由一个关于磁盘文件的内部数码替换它，则后面的所有对标准输出的写操作均写到该磁盘文件去。

如上所述，DOS 允许一次打开文件个数为 20 个，扣除标准设备所占的 5 个，即实际上能一次打开的文件个数为 15 个，如果需要在一次打开超过 15 个用户文件，则需要设置新的文件表，该文件表的长度及指针由 PSP+32~37H 指示。

7) 环境块段址

此域的偏移量为 2CH，占两字节。所谓环境是内存中的一个区域，主要用于存贮由 DOS 的 PATH, SET, PROMPT 命令设置的参数。它由一系列以 00H 结束的 ASCII 串组成。本区域包含该环境的段址。如果该段址值为 0000H，表示为内部环境块（由 COMMAND.COM 作为环境），其值不为 0 则表示为外部环境块。

8) 用户程序堆栈指针

本域偏移量在 2EH，占 4 个字节，存放用户程序堆栈指针 SS: SP。DOS 功能调用主控程序总是将发出功能调用请求的用户程序当前栈指针存放在这 4 个字节中，在用户程序运行结束时，恢复原来的值。

9) 扩展 FCB1*

本域偏移量在 0055~005BH，共 7 个字节，详细参看稍后叙述的 FCB。

10) 命令行/默认的磁盘传送缓冲区

偏移量 80H 处的一字节域开始时存放命令行长度，而命令行存放在偏移量 81H 后，最多包含 127 个字节。此域包含的是无格式的命令行拷贝，以程序名后打入的第一个空格开始，且是该命令行的严格拷贝。但只有一个例外，即重定向符（<、>），管道符（|）以及与其相关的文件名均被命令处理程序删除了。此域的信息对用户程序是透明的。

此区域另一个用途是作为默认的磁盘传送缓冲区（DTA），因此，在调用任何使用到 DTA 的子程序之前先保存命令行长度和命令行。磁盘传送区 DTA 是文件控制块类的命令用于在文件间交换数据的缓冲区，其长度为 128 个字节。

我们知道，在大、中、小型计算机的多用户操作系统中，系统是通过进程控制块（PCB）对进程进行管理的，系统根据 PCB 而感知进程的存在，PCB 是进程唯一的实体。系统创建某一进程时，首先建立了该进程的进程控制块，然后再利用它对进程来进行

控制和管理。在进程运行结束时，系统通过撤消其进程控制块，回收其内存资源，该进程也随之消亡。

MS-DOS 具有 EXEC 的功能，也即一个被装入到用户区的系统程序（如 COMMAND.COM）或用户程序，能够作为父（parent）程序再加载一个用户子（child）程序，而该用户子程序又可以再加载另一个用户子程序，只要当前内存有足够空间，就可连续调用 EXEC 的功能，实现多个子程序从磁盘加载到内存并依次执行。父程序与子程序之间的信息传递就通过各自的程序段前缀 PSP 以及程序环境块来进行，MS-DOS 是单用户单道程序运行的操作系统，其 PSP 的作用就类似于进程控制块的作用，它存放以下四部分信息，供操作系统管理之用：

- ①供被加载的子程序使用的 DOS 入口，如 PSP+0, +2H, +5H, +2CH;
- ②供 DOS 本身使用的 DOS 入口，如 PSP+0AH, +0EH, +12H;
- ③传递给子程序的参数，如 PSP+5CH, +6CH, +80H;
- ④有关设备或文件本身使用的信息如 PSP+18H~2BH, PSP+32~37H。

2. 文件控制块 (FCB)

文件控制块 (FCB) 用于文件操作时存放记载、标识、说明和控制文件的参数。传统的系统调用（即从 CP/M OS 发展过来的 24H 号以下的系统调用）访问文件时要用到 FCB。标准的 FCB 有 37 个字节，其中顺序文件用到 33 个字节，随机文件用到 37 个字节。此外，在建立或搜索磁盘目录中文件属性时还要用到扩展的 FCB (7 个字节)。COMMAND 模块中设置多个 FCB，其中有 PSP 表中首址为 CS: 005CH 和首址为 CS: 006CH 的 FCB1 和 FCB2 (缺省的 FCB，仅用标准 FCB 的前 16 个字节存放文件名参数)；常驻程序中为批处理 FOR 命令后文件参数而建立的 FCB 以及暂驻程序中为命令读取和识别操作而建立的 FCB 等，其结构如表 1.7 所示。

表 1.7 文件控制块 FCB

	-7	FF	属性 (1B)	扩展 FCB
0	盘号 (1B)	文件名 (8B)		标准 FCB
8	文件名	类型名 (3B)	当前块号 (2B)	逻辑记录长度 (2B)
16	文件长度 (4B)	日期 (2B)	时间 (2B)	
24	保留给系统使用			
32	当前记录 (1B)	随机记录号 (4B)		

FCB 各域的意义如下：

- * 盘号：0 表示当前盘（文件打开后改成对应的盘号）；1~4 表示 A~D 盘
- * 当前块号：当前相对块号，每块由 128 个记录组成（用于顺序文件读写）
- * 逻辑记录长度：以字节为单位的逻辑记录长度（使用功能调用前置成 80H）
- * 当前记录：当前块内的当前记录的相对序号（用于顺序文件读写）
- * 随机记录号：当前随机记录号（用于随机文件读写）
- * 属性：含义同目录项的属性（见后），只有当 FCB-7H 为 FF 时才有效
- * 扩展 FCB 可用来生成或访问具有特殊属性（如隐含、系统或只读文件）、卷标、子目录的文件，扩展 FCB 安排在 PSP 中 FCB1 之前，即 0055~005BH 处，且

PSP+0055H (即 FCB-07H) 字节的值置成 FFH, 该扩展 FCB 才能成立。

3. 文件目录项 (FDT)

DOS 采用目录登记的方式对文件进行管理, 当建立一个新文件时, DOS 便将其文件标识及其他特征按系统设定的目录格式登记在一个空的文件目录项上, 也就是说文件目录项提供给 DOS 目录操作用, 为此 DOS 在磁盘一定区域开辟了磁盘目录区, 360KB 软盘的目录区在逻辑扇区 05~0CH, 共 7 个扇区, 1.2M 软盘目录区在逻辑扇区 0FH~1CH 共 14 个扇区, 40M 硬盘根目录区在逻辑扇区 7B~9AH 共 32 个扇区。每个文件目录项占 32 个字节, 故 DOS 允许最大的根目录数 360KB 软盘为 112 个, 1.2MB 软盘为 224 个, 20M 硬盘为 512 个。由于 DOS 采用树型目录结构, 能提供子目录功能, 并将子目录作为文件来处理, 因此, 从理论上来说, MS-DOS 的磁盘文件个数仅受磁盘空间的限制。其结构如表 1.8 所示。

表 1.8 文件目录项

00H		07H		08H		0AH		0BH		0CH		0FH			
文件名				扩展名				属性				保留			
保留		时间		日期				起始簇号				文件长度			
10H	15H	16H	17H	18H	19H	1AH	1BH	1CH	1DH	1EH	1FH	20H	21H		

其中各项意义如下:

* 文件名: 其中第 0 字节表示目录项的状态

00H: 从未用过的目录项

05H: 表示文件名的第 1 个字符是 E5H

E5H: 已删除文件的目录项

2EH: 子目录标记项 (如果下一字节也为 2E, 表示为子目录表的第 2 个登记项, 其起始簇号包含父目录的起始簇号, 如此值为 0, 则表示父目录是根目录)。

其它值: 文件名的首字符

* 属性: 共六种文件属性, 其中有些属性可组合使用, 含义如下 (表 1.9):

表 1.9 文件属性字段

×	×	更改位	子目录	卷标位	系统	隐含	只读
---	---	-----	-----	-----	----	----	----

其中隐含、系统和子目录文件不能用一般的目录操作检索; 只读文件只能读不能写; 卷标位为 1, 表示该目录项记录着这张盘的卷名, 更改位在文件修改后置 1。

* 时间和日期: 记录文件最近一次更改的时间和日期

* 簇号: 文件的首簇号

* 文件长度: 以字节为单位的文件长度

上述这些表格的数据结构可用元素类型为记录的数组来描述。

在时间、日期字段中时间 2 个字节共 16 位表示, 其格式如下:

0~4 位表示秒, 其值为 0~29 (以每 2 秒为 1 个单位);

5~10 位表示分, 其值为 0~59;

11~15 位表示时, 其值为 0~23。

日期也用 2 个字节共 16 位表示, 其格式如下:

0~4 位表示日, 其值为 1~31;

5~8 位表示月, 其值为 1~12;

9~15 位表示年, 其值以 1980 年为基值。

时、分、秒、日、月、年字段均以 16 进制形式来表示。

例如, 我们读出 IBMBIO.OCM 文件的目录项内容是:

```
64CF:0000 49 42 4D 42 49 4F 20 20-43 4F 4D 27 00 00 00 00 IBMBIO COM. . . .
64CF:0010 00 00 00 00 00 00 00 60-71 0E 02 00 54 56 00 00 . . . . ' q. . . TV. .
```

其属性字段值为 27, 即其更改位、系统位、隐含位、只读位值为 1, 该文件为系统文件、隐含文件和只读文件。

其时间值为 6000H, 即:

0~4 位值为 0

5~10 位值为 0

11~15 位值为 01100 即 0CH, 十进制值为 12 时。

其日期值为 0E71H, 即

0~4 位值为 10001, 即 11H, 十进制值为 17。

5~8 位值为 0011, 即 3H, 十进制值为 3。

9~15 位值为 0000111, 即 07H, 十进制值为 7, 加上基值 1980, 故为 1987 年。

也就是说, IBMBIO.COM 文件建立的日期和时间为 1987 年 3 月 17 日 12 时正。

该文件起始簇号为 02 簇, 该文件长度值为 5654H 即 22100HBytes。

4. 文件分配表 (FAT)

为了更好地利用磁盘空间, DOS 为每个文件分配的扇区并不要求一定要连续。为此, DOS 设置了文件分配表来记录每个文件在磁盘扇区的位置。文件在磁盘的分配是以簇为单位的, 不足 1 簇的零头也分配 1 簇。因此 FAT 表实际是就是一个文件在磁盘上分配的簇号表。在 FAT 表中, 每一簇号登记项占用的位数有 12 位 (1.5 个字节) 和 16 位 (2 个字节) 两种, 具体选择根据 DOS 版本及盘容量而定 (如表 1.10 所示)。

表 1.10 FAT 表簇号登记项位数

版本号	2.x	3.x	
位数	12 位	盘容量 < 4087 簇	盘容量 ≥ 4087 簇
		12 位	16 位

FAT 表簇号登记项的结构如表 1.11 所示:

表 1.11 簇号登记项

000	001	002	nnn
保留簇		文件簇号链	

其各簇意义如下:

000~001 簇：保留簇，共 3 个字节（12 位 FAT）或 4 个字节（16 位 FAT）

其中第 1 个字节为介质描述符，与磁盘 I/O 参数表相应描述是一致的（例如 FD 表示为双面软盘，每道 9 个扇区）第 2~3（或 2~4）个字节为 FFH。

002 簇~：为使用簇，记录文件的簇号链，其各值含意如表 1.12 所示

表 1.12 使用簇各值意义

12 位	16 位	意 义
000H	0000H	可用簇
FF0~FF6H	FFF0~FFF6H	备用簇
FF7H	FFF7H	坏簇
FF8~FFFH	FFF8~FFFFH	末簇
001~FEFH	0001~FFEFEH	下一簇号

为了安全起见，在磁盘中设立两个 FAT 表（一个使用，一个备份）。各类规格磁盘的 FAT 表在磁盘中位置如下（表 1.13）

表 1.13 各类磁盘 FAT 表的位置

表项	磁盘规格	360KB 软盘 (PC/XT 机)	1.2M 软盘	20M 硬盘	40M 硬盘
	逻辑扇区号				
FAT		1~4	1~0EH	1~52H	1~7AH
FDA		5~0BH	0F~1CH	53~72H	7B~9AH

那么，如何查找文件在磁盘上分配的簇号呢？下面对 12 位和 16 位 FAT 分作介绍。

1) 12 位 FAT

- ①将当前逻辑簇号乘以 1.5；
- ②其积整数部分作为 FAT 字节位移；
- ③取出以该字节位置开始的一个字的值（注意高低字节位置的读法）；
- ④如当前簇号为偶数，取该字值的低 12 位，如当前簇号为奇数，取该字值的高 12 位；

- ⑤如遇 FFFH，表示为末簇。

2) 16 位 FAT

- ①将当前逻辑簇号乘以 2；
- ②其积作为 FAT 字节位移；
- ③取出以该字节位置开始的一个字的值作为下一簇号；
- ④如遇 FFFFH，表示为末簇。

下面，以 MS-DOS 3.3 版系统盘上（使用 360KB 软盘）FAT 表为例，说明如何根据 FAT 表查找簇号链，从而获知文件在磁盘中存放的位置。

首先，用 DEBUG.COM 读出该版本 FAT 表的内容（为节省篇幅，下面只读出 1 个扇区）：

-lcs:100 0 1 1

-dcs:100 2ff

```

1040:0100 FD FF FF 03 40 00 05 60-00 07 80 00 09 A0 00 0B ....@
1040:0110 C0 00 0D E0 00 0F 00 01-11 20 01 13 40 01 15 60 .....@
1040:0120 01 17 F0 FF 19 A0 01 1B-C0 01 1D E0 01 1F 00 02 .....
1040:0130 21 20 02 23 40 02 25 60-02 27 80 02 29 A0 02 2B ! # @ .% ./.+.
1040:0140 C0 02 2D E0 02 2F 00 03-31 20 03 33 40 03 35 F0 ..- .. / ..! .3@ .5.
1040:0150 FF 37 80 03 39 A0 03 3B-C0 03 3D E0 03 3F 00 04 .7...9...;...=...?...
1040:0160 41 20 04 43 40 04 45 60-04 47 80 04 49 A0 04 4B A .C@ .E .G..I..K
1040:0170 C0 04 4D E0 04 FF 0F 05-FF 2F 05 53 40 05 55 60 ..M...../ .S@ .U
1040:0180 05 57 80 05 59 A0 05 5B-C0 05 5D E0 05 5F 00 06 .W..Y..[.]...
1040:0190 61 20 06 63 40 06 65 60-06 67 80 06 69 A0 06 6B a .c@ .e .g..i..k
1040:01A0 C0 06 6D E0 06 6F 00 07-FF 2F 07 FF 4F 07 75 60 ..m..o.../ ..O..u
1040:01B0 07 77 80 07 79 A0 07 7B-C0 07 7D E0 07 7F 00 08 .w...y...{...}.....
1040:01C0 81 20 08 83 40 08 85 60-08 87 80 08 89 A0 08 8B . .@ .. .....
1040:01D0 C0 08 8D E0 08 8F 00 09-91 20 09 93 40 09 95 60 .....@
1040:01E0 09 FF 8F 09 99 A0 09 9B-C0 09 9D E0 09 9F 00 0A .....
1040:01F0 FF 2F 0A A3 40 0A A5 F0-FF A7 80 0A A9 A0 0A AB ./ ..@ .....
1040:0200 C0 0A AD E0 0A AF 00 0B-B1 20 0B B3 40 0B B5 F0 .....@
1040:0210 FF B7 80 0B B9 A0 0B BB-F0 FF BD E0 0B BF 00 0C .....
1040:0220 C1 20 0C FF 4F 0C C5 60-0C C7 80 0C C9 A0 0C FF , .....O..
1040:0230 CF 0C CD E0 0C CF F0 FF-D1 20 0D D3 40 0D D5 60 .....@
1040:0240 0D D7 80 0D D9 A0 0D FF-FF FF DD E0 0D DF 00 0E .....
1040:0250 E1 20 0E E3 40 0E E5 60-0E E7 80 0E E9 A0 0E EB . .@ .. .....
1040:0260 C0 0E ED E0 0E EF 00 0F-F1 20 0F F3 40 0F F5 60 .....@
1040:0270 0F F7 80 0F F9 A0 0F FB-C0 0F FD E0 0F FF 00 10 .....
1040:0280 01 21 10 03 41 10 05 61-10 07 81 10 09 A1 10 0B !...A...a.....
1040:0290 F1 FF 0D E1 10 FF 0F 11-11 21 11 13 F1 FF FF 6F .....!.....o
1040:02A0 11 17 81 11 19 F1 FF 1B-C1 11 1D E1 11 1F 01 12 .....
1040:02B0 21 21 12 FF 4F 12 25 61-12 27 81 12 29 A1 12 2B !!...O.%a'...)...+
1040:02C0 C1 12 2D E1 12 2F 01 13-31 71 13 F7 7F FF F7 7F ..- .. / ..!q.....
1040:02D0 FF F7 FF FF 39 A1 13 40-71 FF F7 7F FF F7 7F FF ....9..@ .q.....
1040:02E0 FF 2F 14 43 21 15 F7 7F-FF F7 7F FF F7 7F FF F7 ./ ..Cl.....
1040:02F0 7F FF F7 7F FF F7 7F FF-F7 7F FF 53 F1 FF F7 0F .....S.....

```

我们从前面目录区中读出 IBMBIO.COM 文件在文件中的首簇号为 002 簇，根据上面介绍读 12 位 FAT 表的方法，得 IBMBIO.COM 文件在磁盘中分配的簇号链，具体读法如表 1.14 所示。从表 1.14 读得该文件的簇号链是：002 >003 >004 >005 >006。

表 1.14 读 FAT 表示例

文件名	逻辑簇号	FAT 字节位移	以该字节位置开始一个字的值		下一簇号 (取低(高)域值)
			按清单顺序	读作	
IBMBIO.COM	002	3	0340	4003	低 003
	003	4	4000	0040	高 004
	004	6	0560	6005	低 005
	005	7	6000	0060	高 006
	006	9	FF8F	8FFF	低 FFF

二、栈

栈实际上是一种特殊的线性表，其特征是后进先出。这种特性便于保存后面操作用到的数据，以免重新运算增加空间和时间的开销。在程序运行的过程中，若中间需插入其他