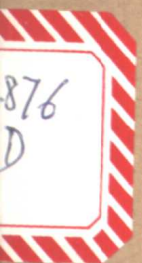


CP/M 微型机磁盘操作系统



CP/M 微型机磁盘操作系统

——结构与使用

- 发展概况
- 使用方法
- 模块结构
- 功能调用
- 操作过程
- 修改移植

科学技术文献出版社重庆分社

CP/M 微型机磁盘操作系统

清华大学计算机系微型计算机研究室
中国科学技术情报研究所重庆分所
科学技术文献出版社重庆分社
重庆市市中区胜利路91号

四川省新华书店重庆发行所
科学技术文献出版社重庆分社印刷厂

开本: 787×1092毫米1/16 印张: 11 字数: 2

1982年4月第一版

1982年4月第一次印

科技新书目: 19-183

印数: 4,200

书号: 13176·104

定价: 1.15



说 明

CP/M (Control Program/Monitor)操作系统是美国Digital Research公司的产品。标准的CP/M版本是用在Intel MDS800微型机开发系统上的。它是目前世界上最流行的微型机磁盘操作系统,享有“软件总线”之称,其软件丰富、用途广泛、可靠灵活、便于移植,为国内外大多数微型机所采用。

本书在分析全部CP/M程序和综合大量有关资料的基础上,全面介绍和深入剖析了CP/M操作系统及其组成、功能结构、支持软件、对硬件的适应性和移植方法;详细叙述了CP/M的内部程序结构、设计方法和执行过程;结合实例介绍了CP/M的使用、修改和移植。附有多个程序框图和清单。收进了全部CP/M软件手册的资料。

本书以实用为目的,力求做到深入浅出,保持资料的系统性和完整性,希望有助于加深对CP/M的了解,有助于CP/M的推广、使用和移植。

本书由清华大学计算机系张公忠、李建桐、戴梅尊等同志编写,由《微型计算机》编辑部林云梯同志通读编排。由于我们水平有限,时间仓促,错误在所难免,恳请读者批评指正。

清华大学计算机系微型计算机研究室
中国科技情报所重庆分所《微型计算机》编辑部

目 录

第一部份 CP/M的结构与执行过程

第一章 概述	(1)
§1.1 CP/M操作系统的发展	(1)
§1.2 CP/M支持的系统软件	(2)
§1.3 CP/M的基本结构和功能	(5)
§1.4 CP/M对硬件的适应性	(9)
§1.5 CP/M的移植	(11)
附录: 采用CP/M的主要机型一览表	(12)
第二章 CCP的结构与执行过程	(13)
§2.1 CCP具有的功能	(13)
§2.2 CCP的结构	(15)
§2.3 CCP的执行过程	(19)
附录1: CCP程序总框图	(21)
2: DIR执行程序流程图	(21)
3: ERA执行程序流程图	(21)
4: REN执行程序流程图	(21)
5: SAVE执行程序流程图	(22)
6: TYPE执行程序流程图	(22)
7: 外部命令处理程序流程图	(22)
第三章 BDOS的结构与执行过程	(23)
§3.1 磁盘和磁盘的基本结构	(23)
§3.2 磁盘的位图与扇区的交叉排列	(23)
§3.3 文件名, 目录与文件控制块	(25)
§3.4 磁盘文件的管理	(31)
§3.5 BDOS的总体结构	(38)
§3.6 系统调用命令	(40)
第四章 BIOS结构	(47)
§4.1 BIOS模块组成	(47)
§4.2 外围控制程序	(47)
§4.3 CP/M的引导	(48)
§4.4 转移矢量表	(50)
第五章 BIOS实例	(51)
§5.1 系统硬件结构概述	(51)
§5.2 系统软件结构	(52)
§5.3 外围控制程序流程图和程序清单	(55)

第二部份 CP/M的使用、调用及移植

第一章 CP/M的特点和使用	(77)
§1.1 引言	(77)
§1.2 CP/M功能概述	(78)
§1.3 接通磁盘操作	(79)
§1.4 内部命令形式	(79)
§1.5 行编辑与输出控制	(81)
§1.6 外部命令	(81)
§1.7 BDOS 的错误信息	(94)
§1.8 在MDS上CP/M的操作	(94)
第二章 文本编辑ED使用手册	(95)
§2.1 ED概述与操作命令	(95)
§2.2 ED的出错情况	(100)
§2.3 控制字符及命令	(101)
第三章 汇编程序ASM使用手册	(102)
§3.1 引言	(102)
§3.2 程序格式	(103)
§3.3 操作数结构	(103)
§3.4 伪操作指令	(105)
§3.5 操作码	(108)
§3.6 错误信息	(111)
第四章 动态调试程序DDT使用手册	(112)
§4.1 引言	(112)
§4.2 DDT命令	(113)
§4.3 执行说明	(116)
§4.4 举例	(117)
附录: 编辑、汇编、动态调试综合操作实例	(117)
第五章 接口与功能调用方法	(127)
§5.1 引言	(127)
§5.2 操作系统调用规则	(128)
§5.3 功能调用举例	(134)
第六章 修改与移植方法	(137)
§6.1 引言	(137)
§6.2 第一级系统重新生成	(137)
§6.3 第二级系统重新生成	(139)
§6.4 GETSYS和PUTSYS程序	(141)
§6.5 软盘结构	(143)
§6.6 BIOS入口	(144)

§6.7 BIOS 程序	(14)
§6.8 冷启动装入程序	(148)
§6.9 零页保留单元	(148)
§6.10 CP/M 1.3版本用户须知	(149)
附录A: MDS装入程序的移动程序清单	(149)
B: MDS冷启动装入程序清单	(150)
C: MDS基本I/O系统(BIOS) 程序清单	(153)
D: 提纲式的CBIOS 程序清单	(162)
E: 提纲式的GETSYS/PUTSYS 程序清单	(167)
F: 提纲式的冷启动装入程序清单	(169)

第一部分 CP/M的结构与执行过程

第一章 概 述

§1.1 CP/M操作系统的发展

CP/M (Control Program/Monitor) 操作系统是美国Digital Research公司为微型机设计的一种通用磁盘操作系统。标准的CP/M版本是用在Intel MDS800微型机开发系统上的。我们知道,操作系统作为应用程序和硬件之间的接口,总是具有一定的服务范围并且依赖于一定的硬件,因而一种操作系统通常只用在同一种机器系列上。CP/M是一种通用控制程序,具有软件丰富、用途广阔、可靠易懂、便于更改和移植等优点,因而被多家厂商采用,成为目前世界上最流行的一种微型机磁盘操作系统。配置CP/M操作系统的微型机种类繁多、数量庞大,其中包括了大多数流行的以8080/8085/Z80为CPU、带有磁盘和各种外设的微型计算机系统。后面列出了使用CP/M操作系统的主要机型。很多微型机原来配有自己的操作系统,如TRS-80等,又加配了CP/M操作系统。另有一些机器配备了和CP/M相兼容的操作系统,如Cromemco上的CDOS,以便享用CP/M支持的软件。

CP/M自1975年问世以来,其版本不断地改进和完善,迄今,比较成熟和流行的版本有:

CP/M1.4 它适用于CPU为8080/8085/Z80、有1至4台单密度软盘的计算机。是1979年以前最完善和最流行的CP/M版本。至今仍然为相当多的微型机所采用。

CP/M2.2 它适用于CPU为Z80/8085、带

有各种类型软盘和硬盘的机器。在CP/M1.4的基础上增加了多用户共享磁盘、磁盘随机存取、文件只读标志和系统标志等许多功能。随着Z80为CPU的机器取得优势地位,配备CP/M2.2的机器越来越多。

MP/M 它是与CP/M兼容的多用户、多作业操作系统。支持每个终端上的实时多道程序,同时运行编辑程序、程序翻译程序和后台打印机假脱机程序,以及远距离数据输入和数据库存取,是CP/M各版本功能最强者。

CP/NET 它是与CP/M相兼容的微型机网络操作系统。允许独立的微型机访问外设、程序和数据等公共设施,是MP/M和CP/M组合在一起运行的系统。主机由MP/M操作,而从机与主机之间的通信由CP/M来操作。应用范围从多终端文字处理/数据库系统到工业过程控制系统。

Digital Research公司一开始就设计出CP/M的十种公用程序。它们是:

ED文本编辑程序, ASM8080汇编程序, LOAD建立可执行文件程序, DDT动态调试程序, STAT文件存贮统计和外设赋值程序, PIP外部交换程序, DUMP代码文件转贮程序, MOVCPM系统重新定位程序, SYSGEN系统生成程序, SUBMIT成批命令处理程序。

随着CP/M的不断推广普及,Microsoft、Micro Focus、Compiler Systems、Micropro、Graham-Dorian Software Systems、Structured Systems Group等软件公司开发了大量的CP/M支持软件。大大提高了

CP/M在微型机领域中的地位。目前,无论是CP/M支持的软件,还是配置CP/M的微型机的种类和数量,都仍然保持着发展扩大的趋势。

CP/M在微型机操作系统中并非功能最强、设计最精巧的一个。它不是一个实时系统,MP/M之前的各个版本都没有分时和多道程序功能。它之所以广为流行,主要有以下原因:

1.CP/M是用Intel 8080指令系统写成的,适用于CPU为8080/8085/Z80的微型机,而这类微型机一直在市场上占优势地位。

2.CP/M采用层次结构,可靠易懂、适应性强,并且具有便于更改、发展和移植的优点。

3.CP/M的功能和结构也便于开发系统软件。

4.已经开发了大量的系统软件和应用软件。其数量之多、内容之广是其它操作系统不能与之相比的。

目前,国内生产的052、054和BCM—1、BCM—2机也已配备上CP/M操作系统。国产的050系列S—100总线结构的微型机都将配备统一标准的CP/M磁盘操作系统。随着国产和进口的配备CP/M操作系统的微型机日益增多,可以预期,对CP/M软件系统的研究、移植、推广和应用必将迅速开展起来。

§1.2 CP/M支持的系统软件

评价一台计算机的性能优劣,不仅要看它的硬件指标,而且要看它的软件配置。一台机器的软件配置的最大限度,取决于它的操作系统已经开发了多少种支持软件。通常,操作系统的支持软件包括三大类:公用程序、语言处理程序和应用程序。公用程序提供文件管理、系统管理和开发程序等各种服务,为用户使用计算机提供各种方便。语言处理程序提供用户使用的高级语言,是用户进行程序设计的主要工具。应用程序是为各种应

用目的设计的专用软件。目前,微型机渗透到社会生活的各个方面,大量的微型机要求配备专用软件,应用程序的开发直接关系到微型机的推广应用。

CP/M操作系统的支持软件极其丰富。据统计,目前各家厂商可以提供的CP/M软件高达1500多种,加上各个用户开发的软件约在3000种以上。因而有“软件总线”之称。下面列出市场上提供的主要软件,按功能分三类进行介绍。

1. 系统工具类

在CP/M的各种公用程序中,几乎一切能够想象出来的用于文件管理和系统管理的程序均已开发。下面仅列举用于程序开发和数据通讯方面的公用程序(前面已介绍过的除外):

MAC	8080宏汇编程序
PASM	Z80宏汇编程序
MACRO—80	8080/Z80浮动汇编程序
XASM—05	6805交叉汇编程序
XASM—09	6809交叉汇编程序
XASM—18	1802交叉汇编程序
XASM—48	MCS—48交叉汇编程序
XASM—68	6800交叉汇编程序
XASM—F8	F8/3870交叉汇编程序
XASM—65	MCS—6500交叉汇编程序
XASM—400	NC—400交叉汇编程序
XASM—51	8051交叉汇编程序
XMACRO—86	8086交叉汇编程序

EDIT	面向字符的文本文件编辑程序		容,但执行速度快3—10倍
EDIT—80	快速随机存取行编辑程序	CBASIC—2	非会话式商用BASIC编译程序和运行解释程序
PMATE	具有整面翻卷和双游标特性的文本编辑程序	S—BASIC	结构BASIC编译程序
DISTEL	8080/Z80反汇编程序	XYBASIC	过程控制和实时应用的扩展BASIC会话式解释程序
DISILOG	Z80/5065反汇编程序		
BUG	Z80会话式机器语言调试程序	KBASIC	具有全部 KISS键检索顺序查找性能的BASIC
RAID	8080/8085实时汇编程序调试程序	FORTTRAN—80	ANSI66FORTRAN(除复数外)加上很多扩展功能
ZDT	Z80监控程序调试程序		
SID	8080符号指令调试程序	COBOL—80	ANSI74标准COBOL第一级加第二级的大部分
ZSID	Z80符号指令调试程序		
Z80开发程序包	由文件行编辑程序、Z80浮动汇编程序和连接装入程序组成	CIS COBOL (标准)	ANSI74标准COBOL第一级,支持第二级部分特性
DESPOOL	后台打印程序	CIS COBOL (压缩)	前者的子集,能用在32K的微型机上
FILETRAN	TRS—80文件转换程序		
IBM/CPM	CPM与IBM数据连接程序	NEVADA COBOL	ANSI74标准COBOL子集,用于32K微型机
BSTAM	CP/M微型机之间通讯程序	PASCAL/M	产生P码的扩展标准PASCAL编译程序
BSTMS	智能终端程序		
2. 高级语言类			
高级语言配备非常齐全,并且每种语言往往有各种功能不同的版本,择其主要者罗列如下:			
BASIC—80	ANSI兼容的会话式扩展BASIC	PASCAL/MT	产生8080代码的标准PASCAL的子集,用于32K微型机
BASIC编译程序	和BASIC—80兼容	PASCAL/MT +	标准PASCAL加上符号调试程

PASCAL/Z	产生Z80代码的编译程序		具有全部 HDBS性能的全网络数据库加多级读/写保护功能
JRT PASCAL	标准PASCAL编译程序和运行解释程序	CONDOR	是DBMS 20系列数据库管理系统,可以简化资料的处理
TINY C	用于结构程序设计技术教学的会话式C语言解释程序	WORDSTAR	为非技术人员设计的面向屏幕的文字处理系统
BDS C编译程序	UNIX系统C语言的子集	DATASTAR	专用控制输入和显示系统
WHITESMITHS C编译程序	和UNIX系统版本7C语言完全一致	MICROSPELL	装有25000个英文单词的标准词典的文本拼写校正程序。可以根据需要增加14000个单词
ALGOL---60	Z80系统使用的块结构语言编译程序	TEX	文本输出格式化程序
APL/V80	用于Z80系统的APL SV.子集,开发程序时间约为其它语言的五分之一	LETTERIGHT	可建立、编辑和打印书信文件的文字处理程序
PL/1—80	开发大型结构程序的通用语言。包括300个以上可调用子程序的库	CBS	可构造的商用会话式数据管理系统
muLISP	微型机人工智能LISP语言	SELECTOR II—C2	数据库处理程序。可建立和维持多键数据库,打印格式化的分类报告
muSIMP	微型机符号和半数字处理程序包	SELECTOR IV	检查和处理记录联机的会话式键存取资料管理系统
		POSTMASTER	发送文件清单维持程序
		MAILINGADDRESS	存储和打印姓名和地址资料。每个记录可定义20个参数
		NAD	会话式发文清单建立和维护程序
		MAILMERGE	由发文清单产生标有姓名的书信文件,打印时允许多个文件中进行数据分类

3.应用软件类

CP/M的应用软件极多,大量的应用于商业领域。下面介绍其比较流行的应用软件。

HDBS	可寻找、增加和删除应用程序中的记录的分级数据库管理系统
MDBS	微数据库管理系统。

ACCOUNTSP	AY-	维护发货票和帐单，
ABLE		完成现金流动分析。
ACCOUNTS	RE-	提供顾客付款资料，
CEIVABLE		产生财务报告
INVENTORY	MANA-	库存管理程序。提
GEMENT		供发出、订出等情况
INVENTORY		记录库存级别、价格、
SYSTEM		来源、销售、期限、
		周转额、记帐且提供
		交易资料报告
PAYROLL		工资管理系统
GENERAL	LED-	提供交易登记、每日
GER		记录、试算表和月终
		决算的通用总帐系统
ANALYST		收集、组织和处理数
		据的规格化数据输入
		和产生报告的系统
FPL		微型机财政计划程序
mu MATH		微型机语言会话式符
		号数学系统
STATPAK		专业统计和概率计算
		的程序库

使用较多的软件还有各种语言的公用程序,文件分类/合并程序,字符串处理程序等。上述程序只是巨大的CP/M软件库中的一小部分。其软件的详细资料可从两个著名的CP/M软件销售者Lifeboat Associates和CP/M Users' Group的产品目录中寻找。

§1.3 CP/M 的基本结构和功能

§1.3.1 层次结构

CP/M操作系统程序采用层次结构,即整个操作系统分成若干个基本程序模块,这些程序模块排成若干层,各层之间只能单方向依赖,不构成循环。因而具有比较好的可靠性、易懂性和适应性。也就是说,整个系统的正确性可以由各层模块的正确性来保证,对整个系统的了解变成对各个局部模块

的了解,对整个系统的修改简化为对各个局部模块的修改。

CP/M共分为四个基本模块,即:

BIOS 基本输入/输出操作系统

BDOS 基本磁盘操作系统

CCP 控制台命令处理程序

TPA 过渡性程序区

各模块的层次关系如图1.01所示:

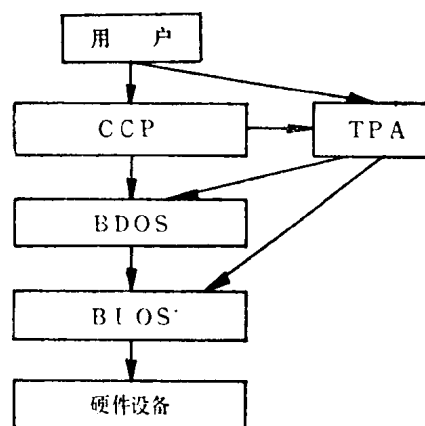


图1.01

BIOS 提供磁盘存取和其它输入/输出设备的主要操作,是CP/M的最底层模块,直接依赖于系统的硬件线路,各种微型机上的CP/M的主要区别在于BIOS,用户也可以根据需要改写它的内容,但必须提供上层模块可以调用的如下15个基本操作子程序:

BOOT	冷启动引导程序
WBOOT	热启动引导程序
CONST	检查控制台状态
CONIN	输入控制台字符
CONOUT	字符输出到控制台
LIST	字符输出到列表设备
PUNCH	字符输出到穿孔设备
READER	输入读带设备字符
HOME	磁头回到磁盘的00道
SELDISK	选择磁盘驱动器
SETTRK	设置磁盘磁道号
SETSEC	设置磁盘扇区号
SETDMA	设置磁盘读写缓冲区指针

READ 读一个扇区到缓冲区
WRITE 把缓冲区 128 个字节写进磁盘扇区

上述子程序中的控制台、列表设备、读带设备和穿孔设备都是指逻辑设备，它们可以规定为相应的物理设备(详见下一节)。

BDOS 主要进行磁盘文件的管理。提供磁盘的文件结构、磁盘空间的分配和文件的存取及外设操作，是CP/M的主要组成部分。BDOS 向上层模块提供一系列的系统调用命令，命令按其功能编号。功能不同的 CP/M 版本1.4, 2.2和MP/M系统调用命令序号分别为0~27 (1BH), 0~36 (24H), 0~36 加 128 (80H)~157 (9DH)。下面列出CP/M版本1.4和2.2的BDOS系统调用命令：

序号	CP/M1.4功能	CP/M2.2功能
0	系统复位	同左
1	控制台输入	同左
2	控制台输出	同左
3	纸带读入设备输入	同左
4	穿孔设备输出	同左
5	列表设备输出	同左
6	(未用)	直接控制台 I/O
7	读I/O字节	同左
8	置I/O字节	同左
9	控制台输出字符串	同左
10	读控制台缓冲区	同左，有改进
11	查控制台状态	同左，
12	校验当前盘	报告版本编号
13	磁盘系统复位	同左
14	选择磁盘驱动器	同左
15	打开文件	同左，有改进
16	关闭文件	同左
17	搜索文件目录项	同左，有改进
18	搜索下一个文件目录项	同左，有改进
19	删除文件	同左，有改进
20	读下一个记录	同左
21	写下一个记录	同左
22	建立文件	同左，有改进

23	文件重新改名	同左，有改进
24	取磁盘登录标记	同左，有改进
25	取当前的磁盘驱动号	同左
26	置DMA缓冲区	
27	查询盘状态	
28	(未用)	磁盘文件写保护
29	(未用)	取R/O向量
30	(未用)	定文件属性
31	(未用)	取磁盘参数地址
32	(未用)	定义用户代号
33	(未用)	磁盘随机读取
34	(未用)	磁盘随机写入
35	(未用)	计算文件大小
36	(未用)	建立随机记录

上层模块在调用这些功能时方法如下：

把命令序号送到C寄存器，入口参数放入寄存器对DE (单字节放入E)，然后执行指令：

CALL 0005H

或JMP 0005H

0005H是BDOS的入口向量。命令执行完毕返回时，一般返回参数存在寄存器A。上述结构使得系统软件的设计工作大为简化，并且便于BDOS功能的修改和扩充。也就是说，系统软件在调用BDOS的不同功能时只需改变命令序号，而不必考虑BDOS内部的地址。反过来说，BDOS内部的变化，除了某些功能改进需要上层调用作相应的修改外，一般原来设计的软件可以不加修改地照样使用。因而给CP/M的发展和系统软件的开发带来了方便。

CCP提供用户控制台和系统之间的接口。负责接受、识别和执行用户打入的键盘命令，命令分为两级：内部命令和外部命令。内部命令即在CCP内部执行，包括列文件目录、删除文件、文件重新命名、内存区作为文件存盘和打印字符文件。外部命令须从磁盘文件中把程序取入内存执行。CCP还通过控制台与用户会话，指示用户进行正确地操作。

TPA存放从磁盘读出的文件。它实际上是可使用的存储器减去CP/M系统区剩下的内存区。一切公用程序，语言程序和用户程序都必须在TPA中运行。CCP和BDOS有时可以被其它程序复盖，因而TPA的大小也是可以改变的。例如DDT调试程序在运行时复盖了CCP，以便给被调试的程序留出更多的空间。如果一个程序复盖了CCP和BDOS或只复盖了CCP，则该程序执行完毕退出时，必须自动引导CP/M操作系统。

§ 1.3.2 内存分配

标准的CP/M操作系统的内存分配如图1.02所示：

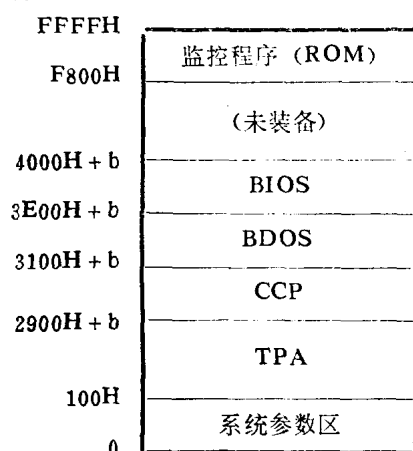


图1.02 标准CP/M内存配图

CP/M可以构成16K至64K各种内存容量的系统。图中系统参数区和监控程序的地址是固定不变的，CP/M各模块的地址随内存容量而改变b的值。当RAM为16K时，b等于零；RAM每增加4K时，b值增加1000H。如48K系统时，CCP开始地址 = 2900H + b = A900H，BDOS开始地址 = B100H，BIOS开始地址 = BE00H。CP/M在系统生成时可根据内存容量自动完成上述模块的地址分配。

必须指出，各种配有CP/M操作系统的微型机，其BIOS和监控程序设计是互不相同的，在驻存方式和地址分配上也就互有区别。图1.02所示为Digital Research标准版本用于MDS800上的地址分配，其BIOS模块调用常驻ROM中的监控程序。而其它机器上的

BIOS有的不调用监控程序；有的BIOS中只保留向量表，执行程序全部放在ROM中；有的只把引导程序放在ROM中，其余在RAM中BIOS模块上；还有的机器采用CRT整体结构，显示缓冲区是内存的一部分。屏幕编辑程序有的在ROM中，有的在RAM中BIOS上，等等。但无论如何，ROM或显示缓冲区必须放在内存的最高地址，以保证系统参数区和CP/M各模块的分配地址。

系统参数区保存系统各模块的入口向量，各模块运行时的重要参数和专用缓冲区。各单元的用途如下：

- 0000H~0002H: 系统热启动向量
- 0003H: IO状态字节
- 0004H: 当前默认的盘号
- 0005H~0007H: BDOS入口向量
- 0008H~003FH: 中断向量 (RST1~RST7)
RST2用于磁盘控制器读写后返回，RST7用于DDT调试程序断点返回，其余保留。
- 0040H~0041H: CP/M的基本地址
- 0042H~0043H: DMA起始地址
- 0044H: 当前选择的盘号
- 0045H: 下一个磁盘操作使用的盘号
- 0046H: 下一个磁盘操作使用的道号
- 0047H: 下一个磁盘操作使用的扇区号
- 0048H: 当前的磁盘类型
- 0049H~004CH: 未选择的磁盘磁头当前位置
- 004DH: A盘的类型
- 005CH~007CH: 文件控制块FCB
- 0080H~00FFH: 磁盘读/写及控制台命令缓冲区

其余单元未用。上述内容是一般情况下CP/M运行时建立的参数，如若用户不需要BDOS时可以重写这些单元。例如一个只包

括控制台I/O的简单程序须在零单元执行时,可以先用CP/M把它装入100H开始的TPA,再用存贮器移动程序把它移到零单元去执行,结束后通过引导程序可以重新建立系统参数。

§1.3.3 软盘结构

CP/M采用标准的IBM 3740软盘格式。每个单密度8英寸软盘分为77个磁道(编号00~76),每个磁道分为26个扇区(编号01~26);每个5¼英寸小盘分为35个磁道(编号00~34),每个磁道分为18个扇区(编号01~18)。每个扇区都规定为128个字节,在BDOS中,磁盘文件的存取操作是以扇区(又称为记录)为单位的。一个盘片在CP/M操作系统下使用时,须先使用FORMAT公用程序使之变成标准的IBM3740格式。

CP/M操作系统放在磁盘的00~01道(指8英寸盘,对于5¼英寸盘则放在00~02道)。这一部分称为系统区,用户不得直接存取,也不能用DIR命令在目录中列出。如果引导程序不是常驻在ROM中,而是存在盘上,规定它必须存在00道01扇区,从02扇区开始是CP/M系统程序。02~76道为文件区,存放公用程序、语言程序、用户程序等一切

系统的或用户的文件。

CP/M的文件结构,是把文件区分成块和记录。1块=8个记录=1024个字节。文件区的总块数为:

$$\frac{(77-2)\text{道} \times 26\text{扇区/道}}{(8\text{扇区/块})} = 243\text{块}$$

块的编号为0~242。文件区分为文件目录区和文件内容区。文件目录区占有0~1块,其余为文件内容区。BDOS把文件内容区以块为单位分配给各个文件,并把块号一一登记在每个文件的目录项中。由于文件空间是动态分配的,一个文件占用的各个块号不一定连续。每一个块内包括的8个记录在逻辑上是连续的,但并不对应于物理上的8个连续扇区。为了减少磁头的移动次数,提高磁盘的读/写速度,逻辑扇区和物理扇区按一定的规则交错排序。BDOS中存有交错排序表供顺序地磁盘扇区读/写操作使用。

文件目录区共有2K字节,分为64个目录项,每项占32个字节,故文件数目最多64个。目录项的前16个字节包括文件属性、文件名、文件类型、文件范围和占用的记录个数等,后16个字节填入文件占用的各块块号。如图1.03所示:

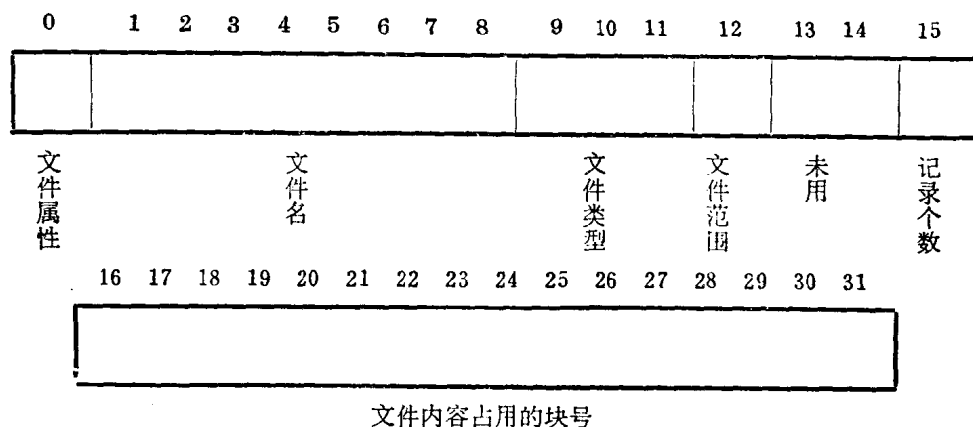


图1.03 文件目录项格式

文件属性: 00H为正常值, E5H表示已被删除。第5、6、7位置1可表示读、写、删除保护。

文件名: 文件名字符,不足8个时补空

格。

文件类型: 文件类型字符,不足3个时补空格。

文件范围: 每个目录项最多16个块号,

称为1个范围。文件内容在16 K以内时范围为0；文件内容在16 K以上 32 K以下范围为1，依次类推。一个文件因而可占多个目录项，文件在盘容量范围内大小不限。

记录个数：文件在本范围内的记录总数，最大值为128个。

占用块号：文件在文件内容区占用的每个块的块号，不足16个时补00H。

CP/M 在进行文件操作时，常常要把文件目录项的内容取到系统参数区的文件控制块FCB，或把文件控制块的内容写到文件目录项中。文件控制块共有33个字节，前32个字节格式和文件目录项一样，只是第1个字节为0~4时分别表文件所在的盘为当前盘、A~D盘，这和文件目录项的文件属性取值不发生矛盾。第33个字节表示下一个磁盘操作的记录号，在进行磁盘读/写时，该记录号每次加1或减1，通过和文件记录个数的比较确定是否读/写完整文件。操作前，通常是把文件控制块第33个字节清零，根据文件名、类型和文件所在盘号，查文件目录项，查到后把记录个数和文件所占块号填入文件控制块相应的单元。操作后，FCB的内容可能会改变，再把前32个字节写回到文件目录区，以保证文件结构的完整性与可靠性。

CP/M 运行时在内存中建立当前磁盘的分配位图，表示磁盘文件区各块的使用情况。每个盘的位图占32个字节，每个字节的8位分别表示8个块，它们是顺序排列的，即第1个字节的0~7位表示第0~7块，第2个字节的0~7位表示第8~15块……。每个字节的某一位为“1”表示相应的块已被文件占用；为“0”表示未被占用。CP/M 在建立文件时，找出未被占用的块号分配给该文件，而在删除文件时，把该文件占用块号对应的位一一清零。

与位图相关的一个字节叫做磁盘登录向

量。它表明CP/M是否已为磁盘建立了位图。该字节的第0、1、2、3位分别对应于A、B、C、D盘。某一位为“1”表示该位对应的盘建立了位图，称为“登录”；为“0”表示没有该盘的位图，称为“登录注销”。热启动可以把磁盘登录向量全部清零，然后为新选择的盘建立位图并登录。因此，在更换盘片时，必须打Control-C键进行热启动。否则，将利用原来盘片的位图作为新盘片的位图分配文件空间，这就会破坏新盘上贮存的文件。

§1.4 CP/M对硬件的适应性

CP/M操作系统对硬件的适应范围和基本要求如下：

CPU为8080/8085/Z80；

8英寸或5 $\frac{1}{4}$ 英寸单密度、单/双面软盘1~4个；

内存容量16K~64K；

外部设备1~16台。

内存容量的改变使用公用程序实现。首先调用系统重新定位程序MOVCPM.COM，在内存中产生你需要的内存容量的CP/M。键盘命令为：

A>MOVCPM $\times \times$ * (cr) 或者 A>MOVCPM * (cr)

第一种命令形式在内存构成 $\times \times$ K（你要求的容量）CP/M，并准备系统生成。第二种命令在内存中构成实际内存容量的CP/M（自动测试系统内存容量），并准备系统生成。然后打入如下命令：

A>SYSGEN(cr)

调用SYSGEN、COM系统生成程序。控制台询问源盘时，打回车键跳过（因为系统已在内存）。当控制台询问目的盘时，即打入你要存放新系统的盘号，再打回车键即可。新操作系统中的地址分配可以符合你改变后的内存容量。

CP/M易于适应各种外设。系统中的

I/O物理设备和逻辑设备的对应关系是通过存放在0003H的I/O字节确定，使用STAT命令可以改变I/O字节的赋值

系统规定有四种逻辑设备：CONSOLE、READER、PUNCH和LIST。I/O字节每2位定义一种I/O设备。格式如图1.04所示：



图1.04 I/O字节格式

每种逻辑设备可以定义1~4种物理设备，因而整个系统最多有16种外设。注意，不同的逻辑设备可以同时定义为一种物理设备，而不同的物理外设不能同时定义为一种逻辑设备。CP/M规定了各逻辑设备允许定义的物理设备，如图1.05所示：

逻辑设备	字节位	取值	定义的物理设备
CONSOLE	0, 1	0	TTY
		1	CRT
		2	READER输入,LIST输出
		3	用户可定义一种设备
READER	2, 3	0	TTY
		1	PTR高速读带机
		2,3	用户可定义二种设备
PUNCH	4, 5	0	TTY
		1	PTP高速穿孔机
		2,3	用户可定义二种设备
LIST	6, 7	0	TTY
		1	CRT
		2	LPT行式打印机
		3	用户可定义一种设备

图 1.05 逻辑设备的定义表

显然，BIOS中应该有这些可定义的物理设备的基本操作子程序，并且能够根据I/O字节的内容，在系统调用逻辑设备基本操作时，转移到规定的相应的物理设备基本操作子程序上去。系统如果有什么外设，BIOS中就应有该外设的基本操作子程序。如若系统

增加了一种外设并且已配备了该外设的硬件接口，但BIOS中没有该种外设的基本操作子程序，则需要对BIOS加以更改。即在BIOS中贴补上该外设的基本操作子程序。贴补的方法如下：

如果BIOS常驻在ROM中，可以调用DDT调试程序，用D命令显示BIOS的机器码和ASCII字符，用L命令印出BIOS的反汇编出来的程序清单，然后把编写好的新增加的设备操作子程序补入原来的清单中，再用EPROM写入设备把贴补后的BIOS程序写入EPROM，安装到机器上开机检查操作是否正确。如不正确则重新修改贴补，直至操作正确。

如果BIOS在盘上，贴补的步骤如下：

1. 调用公用程序MOVCPM，在内存中可以得到浮动的CP/M存储器映象。操作命令为：

```
A>MOVCPM xx * (cr)
```

2. 把浮动的CP/M存储器映象作为文件存于盘上，以便在后里的操作中对该文件进行贴补。

```
A>SAVE 35 CPM xx . COM (cr)
```

这里的35是保存的内存页数，用10进制数表示，每页为256个字节。CPM xx . COM是存盘文件的名称，xx表示内存容量的K字节数。

3. 调用DDT对CPM xx . COM文件进行调试。打入命令：

```
A>DDT CPM xx . COM (cr)
```

用D命令和L命令查看BIOS程序。注意这里的CPM xx . COM是浮动的CP/M存储器映象，其地址和实际的CP/M程序运行时的地址有一个差值n，n的取值与系统内存容量有关：

$$n = 1E80H + b$$

这里的b等于超过16K容量的K字节数。即在16K系统中b=0，32K系统中b=4000H，等等。因此，CPM xx . COM中的程序地址等于CP/M程序运行时的实际地址减去n值。