

INTEL iAPX86,88

CP/M-86

操作系统

李宝山 编译

国营五七四厂

一九八三年六月

INTEL iAPX86, 88

CP/M-86操作系统

李宝山 编译

北京市 2351 信箱 教科印
第二研究室

前 言

CP/M—86 操作系统是 DIGITAL RESEARCH 公司为 INTEL8086 和 8088 十六位微处理机而设计的单用户操作系统。CP/M—86 文件结构同 8080 和 Z80 微处理机使用的 CP/M 操作系统的文件结构兼容。即盘的格式化是相同的。CP/M—86 可以同其它 CP/M 操作系统一样读取数据文件，CP/M—86 的系统调用也可以提供类似的汇编语言编程环境，这就允许用户将其它 CP/M 格式下的应用程序很容易地转换为在 CP/M—86 操作系统下执行的应用程序。

CP/M-86 操作系统的最低硬件环境要求有一个 INTEL8086 或 8088 微处理机的计算机系统，该系统具有至少 32KB 随机存储器，一个键盘控制显示终端、两个 8 吋软盘驱动器及软盘。CP/M—86 操作系统运行自身的应用程序时使用相应的 12KB 内存。为了运行 DDT-86 (Dynamic Debugging Tool) 则需要至少 48KB 内存，而运行 ASM-86 或其它应用程序必须要有 64KB 以上的内存容量。

如果用户的计算机系统比最低硬件环境系统大，则要求 CP/M—86 提供更多的控制功能，这就需要相应的更大一些的内存容量。CP/M—86 可支持 1MB RAM 的内存容量，这是 8086 或 8088 微处理机允许的最大内存。CP/M—86 可支持多达十六个软盘驱动器。

该书简单介绍 CP/M—86 操作系统的系统构成、应用过程及编程调试方法。

由于编译者使用 CP/M—86 操作系统的时间不长、经验不多，加之水平有限，缺点、错误在所难免，望读者批评指正。

编译者 1983.4.11

目 录

前 言

第一章 CP/M—86 操作系统结构

- 1—1 控制台命令处理程序 CCP 及控制台命令..... (1)
- 1—2 基本软盘操作系统 BDOS (5)
- 1—3 基本 I/O 系统 BIOS..... (5)
- 1—4 BIOS 盘定义表..... (11)
- 1—5 CP/M—86 加载和退出过程..... (18)
- 1—6 命令 (CMD) 文件的产生..... (21)

第二章 CP/M—86 操作系统应用

- 2—1 如何启动CP/M—86..... (23)
- 2—2 CP/M—86 命令行编辑控制字符..... (24)
- 2—3 备用文件及系统盘考贝..... (24)
- 2—4 文件、盘、驱动器和设备 (25)
- 2—5 CP/M—86命令概述..... (29)
- 2—6 CP/M—86 操作系统命令..... (31)

第三章 程序编辑与调试

- 3—1 程序编辑..... (48)
- 3—2 程序调试..... (59)

附录：简单 BDOS 调用

第一章 CP/M-86 操作系统结构

CP/M-86 操作系统驻留在一个称之为 CPM.SYS 的文件中,在系统冷启动的初始化过程中该文件被加载到计算机内存中。冷启动加载程序驻留在 CP/M-86 系统盘的头两个磁道中,CPM.SYS 包含三个程序块:控制台命令处理程序块 (CCP)、基本盘操作系统程序块 (BDOS) 和用户构成的基本 I/O 系统程序块 (BIOS),CCP 和 BDOS 两个程序块大约占据 10KB 左右的内存,而 BIOS 的大小随用户构成情况而变。冷启动加载后,所有 CP/M-86 程序块同时被定位并保留在内存中,在热启动时不必重新定位。

CP/M-86 将盘文件加载到计算机内存形成内存映象文件,并且执行这个内存映象文件。内存映象文件的前部有一个“记录头”,它为特定程序的加载和执行提供所需要的信息。在 CP/M-86 操作系统中,内存映象文件由“CMD”文件类型表示。

CP/M-86 系统的应用程序如 ED、PIP、STAT、和 SUBMIT 的应用将在后面的章节中分别叙述。

ASM86 文件用于汇编语言和开发。GENCMD 应用程序用于将 ASM-86 输出的十六进制文件转换为可执行的“CMD”类型文件。LDCOPY 程序用于将盘文件中的冷启动加载程序重新拷贝到磁盘的头两个磁道中去。另外,有一个称之为 LMCMD 的文件,它可以将 INLEL LOC86 应用程序输出的文件转换为 CMD 类型文件,这就大大扩展了 CP/M-86 的功能。还有一个 GENDEF 文件,这是一个产生盘定义文件,它可辅助产生用户盘参数表。

CP/M-86 将内存分为段,每个段最大可到 64KB 连续内存,几个段组成一组。CP/M-86 支持八个段组:代码组、数据组、堆栈组和附加段组,还有四个辅助组。当一个代码组,数据组、堆栈组或附加组被加载时,CP/M-86 分别设置各自的段寄存器 (CS、DS、SS 或 ES) 为该组基地址,CP/M-86 也能加载四个辅助组。一个过程应用程序利用 CP/M-86 存放在用户基页中的数值专门管理辅助组的定位。

1-1 控制台命令处理程序 CCP 及控制台命令

CP/M-86 操作系统被启动加载后,屏幕显示 CP/M-86 的标记信息并在控制台屏幕出现标准提示符。此时,CP/M-86 进入控制台命令处理操作,等待键盘输入命令。这些命令包括:

- DIR
- ERA
- REN
- TYPE
- USER

命令行可以由具有 CMD 文件类型的文件各字开始。

当 CP/M-86 从 CCP 或其它过程应用程序收到加载一个过程应用程序的申请时,它

检查程序所需要的内存是否能够满足，如果内存可以满足要求，则 CP/M-86 分配请求的内存数量给程序并加载该程序，一旦程序加载完成，该程序可以由 BDOS 申请附加的内存用以做为缓冲空间。当程序终止时，CP/M-86 全部释放该程序所占用的程序内存区和附加缓冲区。

CP/M-86 执行过程应用程序分为三种模式：8080 模式、小模式和压缩模式。段寄存器的初始化值由三种模式决定，该值在 .CMD 文件头中描述。

1. 8080 内存模式：

8080 内存模式的代码、数据、堆栈区都在同一组内，段寄存器 CS、DS、ES 均被初始化为组的开始地址，而 SS 和 SP 寄存器在 CCP 内并设置 96 字节堆栈区，指令指针寄存器 IP 设置为 100H。该模式只适于包含一个代码组的过程应用程序使用。程序加载时，8080 模式如图 1-1 所示。

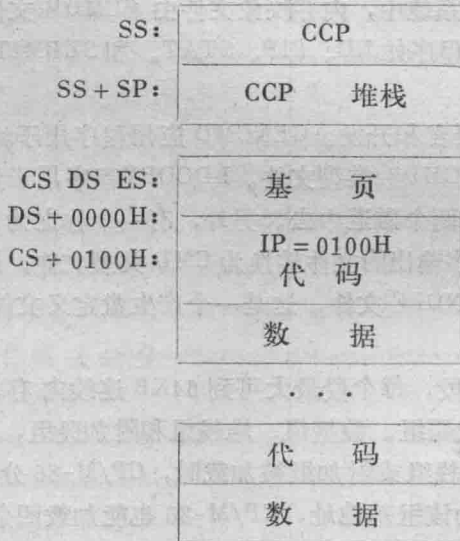


图 1-1 CP/M-86 8080 内存模式

上图中的基页值允许将 8080、8085 或 Z80 代码转换为 8086 和 8088 环境执行的代码。下面是用 ASM-86 编码一个 8080 模式过程应用程序的例子：

```

CSEG
ORG 100H
: (代码)
ENDCS EQU $
DSEG
ORG OFFSET ENDCS
: (数据)
END
    
```

2. 小内存模式：

小模式适用于代码组和数据组分开的程序。在 ASM86 中，这种程序的所有代码紧

跟在 CSEG 指令后面，而所有数据均跟在 DSEG 指令后面，数据独立于代码段。在这种模式中，CS 设置为代码组开始，DS 和 ES 设置为数据组开始，SS 和 SP 保留在 CCP 的堆栈区，如图 1-2 所示。

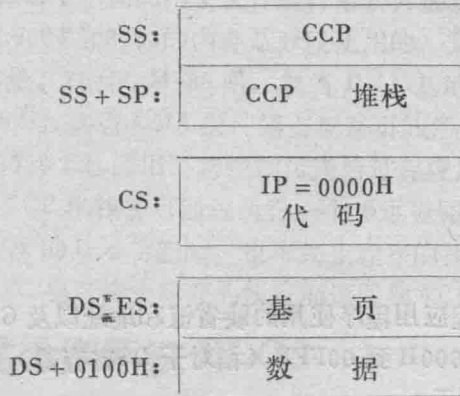


图 1-2 CP/M-86 小内存模式

ASM-86 编码小模式过程应用程序的例子如下：

```

CSEG
: (代码)
DSEG
ORG 100H
: (数据)
END
    
```

3. 压缩内存模式：

当出现单独的代码组、数据组且具有一个或几个保留的堆栈组、附加组或辅助组时使用这种模式。在这种模式下，CS、DS、ES 寄存器设置为各自区域的基地址。图 1-3 说明在压缩模式下段寄存器的初始化情况。

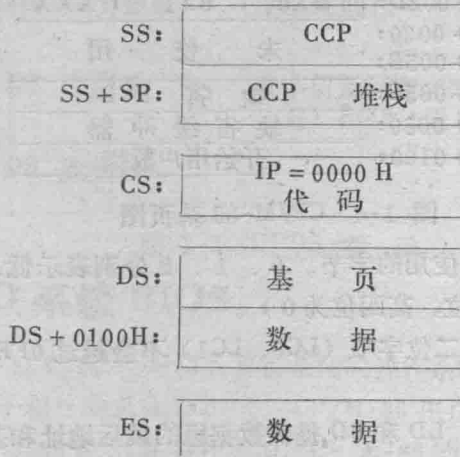


图 1-3 CP/M-86 压缩内存模式

下面用 ASM-86 例程表明如何编码压缩模式应用程序。

```

CSEG
: (代码)
DSEG
ORG 100H
: (数据)
ESEG
: (数据)
SSEG
: (堆栈区)
END
    
```

CP/M-86 基页包含过程应用程序使用的缺省值和地址以及 CCP 初始化的缺省值和地址。基页占据由偏移量 0000H 到 00FFH (相对于 DS 寄存器) 的范围。

基页各值如图 1-4 所示。

DS + 0000:	LC0	LC1	LC2
DS + 0003:	BC0	BC1	M80
DS + 0006:	LD0	LD1	LD2
DS + 0009:	BD0	BD1	XXX
DS + 000C:	LE0	LE1	LE2
DS + 000F:	BE0	BE1	XXX
DS + 0012:	LS0	LS1	LS2
DS + 0015:	BS0	BS1	XXX
DS + 0018:	LX0	LX1	LX2
DS + 001B:	BX0	BX1	XXX
DS + 001E:	LX0	LX1	LX2
DS + 0021:	BX0	BX1	XXX
DS + 0024:	LX0	LX1	LX2
DS + 0027:	BX0	BX1	XXX
DS + 002A:	LX0	LX1	LX2
DS + 002D:	BX0	BX1	XXX
DS + 0030:	未 使 用		
DS + 005B:	未 使 用		
DS + 005C:	缺 省 FCB		
DS + 0080:	缺 省 缓 冲 器		
DS + 0100:	开 始 用 户 数 据		

图 1-4 CP/M-86 基页图

上图中, $\times\times\times$ 表示未使用的字节。0、1、2 分别表示低、中、高字节。LC 是最后代码组的位置 (共 24 位, 高四位为 0)。

在 8080 模式中, LC 低二位字节 (LC0、LC1) 不会越过 0FFFH, 高位字节 LC2 为 0。

BC 是代码组的基地址, LD 和 BD 提供数据组的最后地址和基地址。

当使用 8080 内存模式时, M80 字节等于 1。LE 和 BE 提供选择的附加组的长度和

基地址，而 LS 和 BS 给出选择的堆栈组长度和基地址。字节符号 LX 和 BX 相应设置 4 个选择的独立组，它用于执行压缩内存模式的程序。以上这些字节的初始化值由内存映像文件的记录头给出。

CCP 可以对跟在命令后面的两个文件进行解析并放置格式化的 FCB 在基页的 005CH 和 006CH 位置。由于 8086 和 8088 的内存是分段使用的，其 DMA 地址分为两部分：DMA 段地址和 DMA 偏移量。在 CP/M-86 中，缺省 DMA 基地址初始化为 DS 值，缺省 DMA 偏移量初始化为 0080H，缺省 DMA 缓冲器占据基页的后半部。

CCP 通过一个 8086 “远调用” 将控制传递给过程应用程序，过程应用程序可以选择利用 96 个字节的 CCP 堆栈并可通过执行一个“远返回”终止程序的执行而直接返回到 CCP。当执行 BDOS 的 0 号功能时，也可终止程序的执行。当使用行编辑命令 ED 时，打入一个 CTRL/C 也可终止程序的执行而返回到 CCP 状态。

1-2 基本软盘操作系统 BDOS

CP/M-86 BDOS 共有五十多种基本功能。进入 BDOS 操作是通过 8086 软件中 224 号中断来实现的。中断 224 是 INTEL 公司为 CP/M-86 和 MP/M-86 而保留使用的。功能码在寄存器 CL 中传递，字节参数放在 DL 中，字参数放在 DX 中。单字节值返回在 AL 中，字值返回在 AX 和 BX 中，双字值在 ES 和 BX 中。所有段寄存器，除了 ES，都在进入 BDOS 时被存贮，在退出 BDOS 时重新恢复。图 1-5 总结了输入输出各参数。

CP/M-86 BDOS 功能调用列表在图 1-6 中。BDOS 的各功能在此不一一详述。附录中列出其调用过程。

BDOS 入口寄存器	BDOS 返回寄存器
CL 功能码	字节值返回在 AL 中
DL 字节参数	字值返回在 AX 和 BX 中
DX 字参数	双字值返回在 BX (偏移量) 和 ES (段) 中
DS 数据段	

图 1-5 BDOS 参 数

1-3 基本 I/O 系统 BIOS

CP/M-86 是用于 8086 或 8088 微计算机系统的操作系统，所有硬件特性都集中在各个子程序中，这些子程序的总称为 BIOS。CP/M-86 用户可以修改这些子程序，以使 CP/M-86 适应于任何 8086 或 8088 操作环境。CP/M-86 的 BIOS 驻留在操作系统的最上部，即最高地址部，通常格式如图 1-7 所示。

F#	结 果	F#	结 果
0	系统复位	25	返回当前盘
1	控制台输入	26	设置 DMA 地址
2	控制台输出	27	得到地址
3	阅读器输入	28	写保护盘
4	穿孔机输出	29	得到地址 (R/O 向量)
5	列表输出	30	设置文件属性
6	直接控制台 I/O	31	得到地址 (盘参数)
7	得到 I/O 字节	32	设置/得到用户号
8	设置 I/O 字节	33	随机读
9	打印字符串	34	随机写
10	读控制台缓冲器	35	计算文件大小
11	得到控制台状态	36	设置随机记录
12	返回版本号	37	复位驱动器
13	复位盘系统	40	用 0 填充随机写
14	选择盘	50	直接 BIOS 调用
15	打开文件	51	设置 DMA 段基
16	闭合文件	52	得到 DMA 段基
17	搜寻第一个目录符	53	得到最大内存
18	搜寻下一个目录符	54	得到绝对地址最大内存
19	删除文件	55	得到内存区
20	顺序读	56	得到绝对内存区
21	顺序写	57	释放内存区
22	设置文件	58	释放所有内存
23	重新命名文件	59	程序加载
24	返回逻辑向量		

图 1-6 CP/M-86 BDOS 功能

CS,DS,ES,SS:	控制台命令处理程序 和 基本盘操作系统
CS + 2500H:	BIOS 跳转向量
CS + 253FH:	BIOS 入口点
BIOS:	盘 参 数 表
	未初始化 RAM

图 1-7 CP/M-86 一般结构图

CCP 和 BDOS 由 CP/M-86 的十六进制格式文件 CPM.H86 来提供。为了在非标准硬件上采用 CP/M-86, 必须建立一个 BIOS, 它形成一定的功能, 并将 BIOS 的十六进制文件连接到 CPM.H86 文件的尾部, 然后用 GENCMD 应用程序产生 CPM.SYS 文件以便在冷加载时顺序加载。加载 CPM.SYS 文件到内存去的冷启动加载程序包含一个 BIOS 的简化格式, 称之为 LDBIOS。冷启动加载程序加载 CPM.SYS 到内存中去, 而内存定位地址定义在 CPM.SYS 的头部 (通常地址是 0400H)。

BIOS 是通过一个定位在偏移地址为 2500H 的“跳转向量”来输入的。跳转向量共有 21 个三字节跳转指令, 它将程序控制传送到各个 BIOS 入口点。虽然某些 BIOS 子程序可能只包含一个单独的返回指令 (RET), 但这些子程序的入口点仍可保留在 BIOS 中。表 1-8 列出了各个跳转向量的偏移地址、指令格式、序号和意义。

在 BIOS 中, 各子程序的参数可以在 CX 和 DX 寄存器中传送, CX 接收第一个参数, DX 用于第二个变量。传输返回的值按照这样的规则放在寄存器中: 字节值返回在 AL 中, 字值返回在 BX 中。指定的参数和返回值由每一个子程序来描述。

在 BIOS 跳转向量表中有三个主要的部分:

- 系统初始化子程序。
- 简单字符 I/O 子程序。
- 盘 I/O 子程序。

表 1-8 BIOS 跳转向量

由 BIOS 开始的偏移量	指 令	顺号	意 义
2500H	JMP INIT	0	冷自举
2503H	JMP WBOOT	1	热启动
2506H	JMP CONST	2	控制台字符读状态
2509H	JMP CONIN	3	读控制台字符
250CH	JMP CONOUT	4	控制台字符输出
250FH	JMP LIST	5	列表字符输出
2512H	JMP PUNCH	6	写字符到穿孔设备
2515H	JMP READER	7	读读入设备
2518H	JMP HOME	8	返回磁道 0
251BH	JMP SELDSK	9	选择盘驱动器
251EH	JMP SETTRK	10	设置磁道号
2521H	JMP SETSEC	11	设置扇区号

续表

由 BIOS 开始的偏移量	指 令	顺号	意 义
2524H	JMP SETDMA	12	设置 DMA 偏移地址
2527H	JMP READ	13	读选择的扇区
252AH	JMP WRITE	14	写选择的扇区
252DH	JMP LISTST	15	返回列表状态
2530H	JMP SECTAN	16	扇区译码
2533H	JMP SETDMAB	17	设置 DMA 段地址
2536H	JMP GETSEGB	18	得到 MEM DESC 表偏移量
2539H	JMP GETIOB	19	得到 I/O 字节
253CH	JMP SETIOB	20	设置 I/O 字节

所有简单的字符 I/O 操作都用 ASC II 格式，大写或小写均可，最高位（奇偶校验位）设置为 0。输入设备的文件结束条件由 ASC II 的 CTRL/Z (1AH) 提供。外围设备与 CP/M-86 中的逻辑设备名相互对应。表 1—9 列出了各设备及其特点。

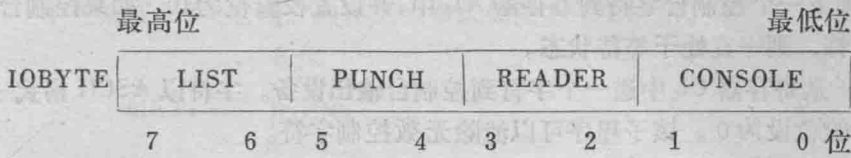
表 1-9 CP/M-86 逻辑设备特点

设备名	特 点
CONSOLE	外设交互控制台，它同操作员通讯，通过 CONST、CONIN、CONOUT 进行存取。
LIST	列表设备。通常是一个硬拷贝设备，例如行式打印机或电传机。
PUNCM	穿带外设。通常是一个高速纸带穿孔机或电传机。
READER	读带设备。例如读入机或电传机。

单个外设可同时设为 LIST、PUNCH、READER 设备。PUNCH 和 LIST 子程序可以简单地返回，READER 子程序可用在寄存器 AL 中的 1AH (CTRL/Z) 返回。

为增加机动性，用户可以选择采用“IOBYTE”功能，它允许重新按装物理设备和逻辑设备。IOBYTE 建立一个对应于物理设备的逻辑表。IOBYTE 是在 BIOS 中保留的一个字节，它对物理设备定义逻辑名。IOBYTE 分为四个性质不同的区域，每区域占两

位，分别称之为 CONSOLE、READER、PUNCH、LIST 区域。IOBYTE 格式如下所示：



每个区域的值可以是 0 ~ 3，用以按装每一逻辑设备的源或目标。可以按排给每个区域的数值及意义列在表 1-10 中。

表 1-10 IOBYTE 区域定义表

CONSOLE 区 (位 0,1)	
0	——控制台为电传机 (TTY:)
1	——控制台为 CRT 设备 (CRT:)
2	——批模式：用 READER 作控制台输入，LIST 作为控制台输出(BAT:)
3	——用户定义控制台设备 (VC1:)
READER 区 (位 2,3)	
0	——READER 是电传机 (TTY:)
1	——READER 是高速读入机 (RDR:)
2	——用户定义读入机*1 (UR1:)
3	——用户定义读入机*2 (UR2:)
PUNCH 区 (位 4,5)	
0	——PUNCH 是电传机 (TTY:)
1	——PUNCH 是高速穿孔机 (PUN:)
2	——用户定义穿孔机*1 (UP1:)
3	——用户定义穿孔机*2 (UP2:)
LIST 区 (位 6,7)	
0	——LIST 是电传机 (TTY:)
1	——LIST 是 CRT 设备 (CRT:)
2	——LIST 是行式打印机 (LPT:)
3	——用户定义列表设备 (UL1:)

BIOS 各个子程序的名称及简单说明如下：

INIT: 在 CPM.SYS 文件被加载到内存之后，CP/M-86 加载程序直接调用这个子程序。

这个子程序初始化所有未经自举加载程序初始化的硬件，设置 BIOS 变量（包括 IOBYTE）的初始化值，并初始化 BDOS 偏移量 (OB11H) 和基地址，送出符号信息。这个子程序完成时，控制转到 CCP，所有段寄存器同时被初始化。

WBOOT: 由 BDOS 功能 *0 调用形成程序终止时调用这个子程序。此时，某些硬件和软件要重新初始化。该程序完成时，控制直接跳到 CCP 热启动入口点 (06H)。

CONST: 接收当前控制台状态并在字符准备读入时在寄存器 AL 中返回 0FFH 值。如果控制台不能进行字符传送则 AL 中返回 00H 值。

CONIN: 读下一个控制台字符到寄存器 AL 中,并设置校验位为 0。如果控制台不输入字符,则一直处于等待状态。

CONOUT: 从寄存器 CL 中送一个字符到控制台输出设备。字符以 ASCII 格式传送。校验位设为 0。该子程序可以滤除无效控制字符。

LIST: 从 CL 送字符到当前列表设备,字符是 ASCII 格式并具有 0 校验。

PUNCH: 从 CL 送字符到当前穿孔机,字符是 ASCII 格式并具有 0 校验。

READER: 从当前输入设备读下一个字符到 AL 中。文件结束由 ASCII CTRL/Z(1AH)表示。

HOME: 返回到当前选中的盘的 00 磁道位置。

SELDSK: 选择 CL 中给定的驱动器。如果是驱动器 A,则 CL 中包含 0,驱动器 B 时,CL 为 1,……驱动器为 P 时 CL 中为 15。当驱动器选中时,SELDSK 在 BX 中返回选中的驱动器盘参数头的基地址。如果企图选择一个不存在的驱动器,SELDSK 在 BX 中返回 0000H 做为错误标识符。

SETTRK: 寄存器 CX 包含磁道号,为选中的驱动器盘的后续存取做准备,可在此时搜索选中的磁道或延迟搜索,直到下一个读/写发生。CL 的值可以在 0~76 范围内,相应于标准软盘磁道号。

SETSEC: 寄存器 CL 包含扇区号。可以在此时将 CL 信息送到控制器或者延迟扇区选择直到读/写操作发生。

SETDMA: 为后续读/写做准备,CL 中包含 DMA(盘内存存取)偏移量。例如,当调用 SETDMA 时,如果 CX = 80H,那么所有后续读操作从当前 DMA 段基址读数据到 80H—0FFH 偏移量。

READ: 假定驱动器已被选中,磁道、扇区也已设置,且 DMA 偏移量和段基址已经指定,READ 子程序读基于这些参数的一个扇区。并返回下列值在 AL 中。

0 —— 盘操作是成功的。

1 —— 发生错误。此时 CBIOS 重复 10 次,看错误是否可以消除。如不能消除,则 BDOS 发出信息:“BDOS ERR ON x: BAD SECTOR”。操作员可用回车忽略错误或用 CTRL/C 返回系统。

WRITE: 从当前选中的 DMA 缓冲器写数据到选中的驱动器、磁道、扇区中。

LISTST: 返回列表设备的准备状态。如果列表设备没有准备好则 AL 中返回值为 00,如果字符可以送到打印机则 AL 中值为 0FFH。

SECTRAN: 形成对物理扇区的译码逻辑。标准 CP/M-86 系统具有“偏移系数”6,在后续读/写操作时,间隔 5 个物理扇区,这个偏移系数在扇区之间允许有足够的时间使程序加载它们的缓冲器,使下一扇区不致丢失。在快速处理器计算机系统,内存和盘子系统、偏移系数可以改变以改进响应速度。

SETDMAB: 在 CX 中放置段基址以为后续 DMA 读/写操作作准备。在读/写操作时,BIOS 将使用由 DMA 基址和 DMA 偏移量决定的 128 个字节的内存缓冲器。

SETSEGB: 在 BX 中返回内存表 (MRT) 地址, 返回值是表的偏移量。该表定义程序提供的地址和物理内存大小。为中断向量和 CP/M-86 操作系统保留的内存区不包括在 MRT 之内。内存区表格式如图 1-11 所示。

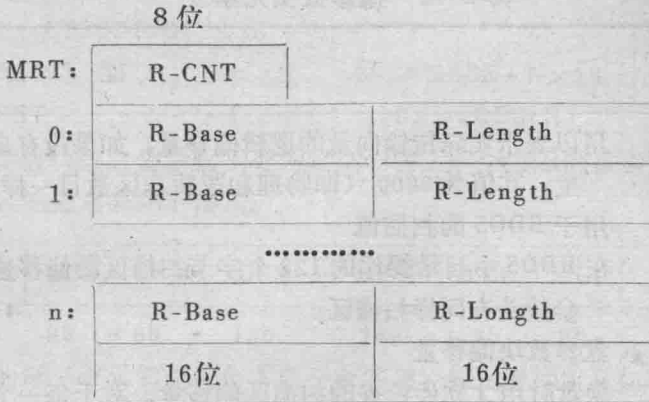


图 1-11 MRT 格式

R-CNT 是内存区描述符的数目 (图中等于 $n+1$)

R-Base 和 R-Length 给出每一物理连续内存区基地址和长度。保留的中断地址 (通常是 0-3FFH) 和 CP/M-86 操作系统不包含在 MRT 中。这里只包含对过程应用程序有用的内存区。如果所有内存区是连续的, R-CNT 为 1 且 $n=0$, 只有一个定义该内存区域的描述符。

GETIOB: 在 AL 中返回 IOBYTE 的当前值。

SETIOB: 用 CL 中的值设置 IOBYTE 值并存放在 BIOS 中。

1-4 BIOS 盘定义表

CP/M-86 是一个表驱动操作系统, 它具有可分区连接的 BIOS。通过改变 BIOS 中的某些子程序, CP/M-86 可以做为在任何 8086 或 8088 微机系统中运行的操作系统。这里我们简单讨论一下在 BIOS 中的一些表的组织和结构。

通常, 每一个盘驱动器都有一个与之相关的盘参数头 (16字节), 它既包含关于盘驱动器的信息, 也提供某些 BDOS 操作的扫描区域。

每个驱动器的盘参数头格式如下所示。

盘 参 数 头							
XLT	0000	0000	0000	DIR BUF	DPB	CSV	ALV
16b	16b	16b	16b	16b	16b	16b	16b

表中每一个元素为一个字 (16位), 每一个盘参数头元素的意义由表 1-12 给出。

如果给出 n 个盘驱动器, 将这些盘驱动器的盘参数头 (DPH) 列成一个表, 表的

第一行，即表的前十六个字节相应于驱动器 0，表的最后一行相应于驱动器 n-1。那么，这个表的格式为表 1-13 所示。

表 1-12 盘参数头元素表

元 素	功 能
XLT	用以表示实际传输向量的逻辑偏移量，如果没有扇区传输发生，其值为 0000（即物理和逻辑扇区数目一样。）
0000	用于 BDOS 的扫描值
DIRBUF	在 BDOS 中目录操作的 128 个字节扫描区的偏移量，所有盘参数头有同样扫描区。
DPB	盘参数块偏移量
CSV	换盘时用于软件检查的扫描区偏移量。对于每一个盘参数头来说，这个偏移量是不同的
ALV	BDOS 用以保持盘定位信息的扫描区的偏移量。这个偏移量对每个盘参数头是不同的。

表 1-13 各驱动器盘参数头表

DPBASE

00	XLT00	0000	0000	0000	DIRBUF	DPB 00	CSV 00	ALV 00
01	XLT01	0000	0000	0000	DIRBUF	DPB 01	CSV 01	ALV 01
.....								
n-1	XLTn-1	0000	0000	0000	DIRBUF	DPBn-1	CSVn-1	ALVn-1

表 DPBASE 定义盘参数头表相对于操作系统开始处的偏移量。

SELDSK 子程序的作用是在操作系统开始时返回盘参数头的偏移量以便选择驱动器。

下面的操作过程返回表偏移量。如果选择的驱动器不存在则返回 0000H 值。

NDISK EQU 4 ; 驱动器数目

.....

SELDSK:

 ; 选择由 CL 给出的盘 N

 MOV BX, 0000H ; 为出错作准备

 CMP CL, NDISKS ; N 超出最大盘号吗?

 JNB RETURN ; 如果是，则返回。

 MOV CH, 0 ;

MOV BX, CX ; BX = N
 MOV CL, 4 ; 为×16作准备
 SHL BX, CL ; N = N×16
 MOV CX, OFFSET DPBASE
 ADD BX, CX ; DPBASE + N×16
 RETURN; RET ; BX ← DPB(N)

驱动器的盘参数块较之盘参数头更为复杂一些。一个实际的盘参数块 (DPB) 是由盘参数头编址的, 它通常有如下形式:

SPT	BSH	BLM	EXM	DSM	DRM	AL0	AL1	CKS	OFF
16b	8b	8b	8b	16b	16b	8b	8b	16b	16b

盘参数块中的每一个元素是一个字或一个字节。各元素的定义如表 1-14 所示。

表 1-14 盘参数块各元素

元 素	功 能
SPT	每磁道中的扇区数
BSH	数据块偏移系数, 由数据块大小决定
BLM	由数据块大小决定的块框架
EXM	由数据块大小和盘上块数决定的范围限度
DSM	决定盘的总的存贮能力
DRM	决定能存在盘上的总目录条目数
AL0, AL1	决定保留的目录块
CKS	目录检查向量大小
OFF	在盘开始处保留的磁道数

表中各元素的值是由 GENDEF 文件自动产生的。BSH 和 BLM 值决定数据块大小 BLS, BLS 不是盘参数块的一个元素。给定一个 BLS 值, 则相应的 BSH 和 BLM 值如表 1-15 所示。

表 1-15 BLS 与 BSH、BLM 对应值表

BLS	BSH	BLM
1024	3	7
2048	4	15
4096	5	31
8192	6	63
16384	7	127