

16位微型机软件资料选编

北京科学技术开发交流中心软件开发一部编

一九八四年六月

编 者 的 话

为了促进国内的软件开发工作,使广大微型机用户有一个良好的开发环境,我们根据最近从国外直接引进的一整套原版软件编写成这个专集,介绍给大家。这套软件全部是受8086/8088型微机系统和CPM-86或MPM-86操作系统所支持的,是目前国内还不多见的最新版本。我们希望这份资料对于正在或即将在这一领域从事软件开发工作的同志们能有一定的参考价值。

MPM-86是目前已被16位微机系统广泛采用的多用户实时操作系统,据不完全统计,已有17种以上比较著名的微机配备了MPM-86。MPM-86与目前国际上已普遍流行的CPM-操作系统完全兼容,有非常丰富的支持软件。我们在本集的开头介绍这个操作系统,目的是为了给后面的讨论提供一个基础。PASCAL/MT+86是DR公司为16位机新设计的完整的PASCAL语言编译系统,它附带一个庞大的子程序库,与标准PASCAL相比,在功能上作了多方面的扩充。CB-86是三次扫描式编译BASIC语言系统,它既保留了Basic原有的特点、但执行速度提高了将近6~10倍。Whitesmith公司的C-86是目前国际上公认的“C”的标准版本,具有对硬件适应性好,占用内存小、执行速度高等独特优点;CIS-COBOL是微型机化的COBOL、它结构紧凑、使用简便,但保留了COBOL语言在数据处理方面的特长;PL/I是从大中型机上移植到微型机的功能较全面的编程语言,它综合吸收了ALGO、LISP和FORTRAN等语言的优点;Wordstar是一个通用性很强的文字处理系统,有数十种可供选择的功能选单,是编写各种公文和信件的理想助手;Supereale是表格处理系统,它能根据输入数据进行自动计算、统计和造表,在国外各商业和事务系统中应用十分普遍。以上八篇文章都是根据原版手册,结合我们的实践与理解,并参照其它有关资料编写而成的。此外,在本集的最后有两篇译文,介绍了完整的信息管理系统和一个先进的数据库管理系统。

去年我们从国外引进了五台ALLOS 8000 16位微机开发系统,有机会对有关16位机的软件资料进行熟悉和消化,这个集子就是在这基础上由几位同志分头编写的,目的是为了与国内的有关单位与工作人员进行技术上的交流。在编写时,我们假定读者是熟悉微型机的或对微型机已有一定了解的,并对各种语言是不陌生的,因此,为了节省篇幅,略去了一般语法书上已经有的基本语法解释和说明,只把重点放在差别和特点、运行所需的软硬件支持环境以及实际使用等方面。

由于时间仓促,实践也不够充分,加之本身水平有限、错误与不妥之处在所难免,若能给予批评指正,我们将不胜感激!

目 录

MP/M-86多用户操作系统	1
PASCAL语言编译系统——PASCAL/MT + 86	32
编译BASIC——CB-86	46
十六位微机的主力语言——WSL-C	59
PL/I-86	68
交互式COBOL语言编译系统——CISCOBOL	79
文字处理系统——WOKDSTAR	87
表格处理程序——SUPERCAL	110
完整的信息管理系统	133
ULTIMATE数据库管理系统	138

MP/M-86 多用户操作系统

沈 为 公

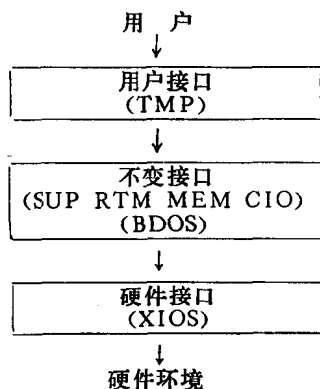
一、概 述

MPM-86是 Digital Reaserch 公司为 8086, 8088系列的16位微型机设计的多用户通用实时操作系统, 这个系统很易于适应各种特定的硬件环境。它最多能支持 254 个字符级I/O 设备(包括控制台和列表机), 每个终端可运行多道作业。MPM-86 最基本的硬件环境包括一个 8086 或类似的CPU、64K RAM、系统控制台和实时钟。一个典型的MPM-86 核心占据大约 32K 字节。MPM-86 兼容单用户操作系统 CPM-86。所有的系统功能均向用户开放, MPM-86也支持各独立进程间的相互通信和同步。

在 MPM-86下, 程序和进程是两个不同的概念, 程序是存放在盘上或内存中的一段代码, 是静态的。进程则是程序运行的轨迹, 是动态的。可以把进程视为执行程序代码和系统代码的逻辑机。MPM-86在装入一个程序的同时也创建了与该程序相对应的进程。实际上是进程控制所有对系统资源的访问, 操作系统的监控对象是进程、而不是程序。

在MPM-86下运行的进程分两大类: 过渡进程和系统进程(包括驻留系统进程)。过渡进程的代码由系统从磁盘装入指定的 TPA 区执行由用户规定的任务。系统进程只执行系统规定的任务, 例如IDLE 进程是一个让系统空转的系统进程。在系统生成时把一部分系统进程并入系统, 使之成为系统的一部分, 称这部分进程为驻留系统进程(RSP)。例如终端消息进程(TMP)是处理控制台命令的RSP。用户也可以生成 RSP。所有在MPM-86下运行的进程使用CPU或其它系统资源时, 均按优先级原则受实时监控模块(RTM)的控制。

MPM-86按功能划分成下列模块: 管理(SUP)、实时监控(RTM)、存贮管理(MEM)、字符I/O(CIO)、磁盘操作系统(BDOS)、扩展I/O 系统(XIOS)。整个系统与外部有三层接口, 用户接口, 不变的逻辑接口和硬件接口。用户接口是通过终端消息进程(TMP)实现的。TMP接受用户命令后激活过渡进程, 或送消息给驻留进程。



MP/M接口示意图

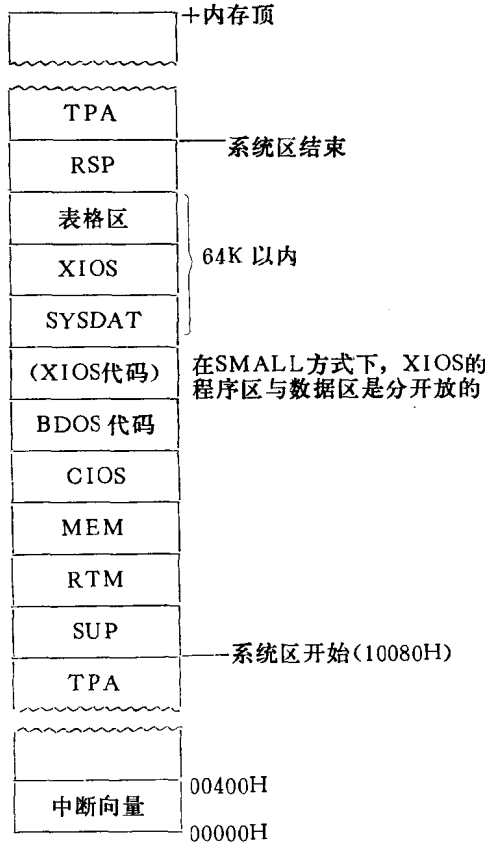
不变的逻辑接口就是指系统功能调用，MPM-86 共有约100种功能调用。这部分除了与系统接口外，也通过硬件与驻留进程和用户进程接口。

硬件接口直接与给定的硬件环境通信。它包括一系列同硬件有关的功能调用。以上三层接口之间的关系如图所示。

图中各代码块包括 SUP 和XIOS均共享一个公共的系统数据区(SYSDAT)。SYSDAT的开始部分的数据结构是固定的，供所有代码块公用。在它后面才是各模块自己的数据区。XIOS的数据区排在最后，在XIOS数据区之后是SYSDAT的表格区、这些表格的大小不是固定的，取决于系统生成时选择的参数。

驻留进程(RSP)被单独安放在紧接SYSDAT之后。系统数据结构，如进程说明(PD)、用户数据区(UDA)和队列结构均位于各自的数据区内。如果这些数据结构位于以 SYSDAT 开始的64K之内，系统将直接使用它们。位于 64K 之外的部分由系统将它们拷贝到前面提到的系统表格区。所有的过渡程序被装到过渡程序区(TPA)、TPA的位置由系统生成时决定。

MPM-86各模块在内存中的位置如下图所示。除中断向量外，整个系统可以位于任意内存区内。各模块的实际地址取决于GENSYS的选择和模块的大小。GENSYS把各模块的地址置入SYSDAT中以供调试检查。XIOS在内存中的安排取决于所选的存贮模式。如果XIOS是以8080模式建立的，则其程序和数据码段合在一起位于SYSDAT区内。如果是以SMALL 模式建立的XIOS，则其程序部分位于SYSDAT区之外，而数据部分仍位于SYSDAT区内。(参看8.3节)



MP/M-86存贮区配置

二、系统管理(SUP)

SUP 负责管理用户程序、系统模块、XIOS 以及未来的网络接口之间的相互联系。所有的系统功能调用均通过SYSDAT中的SUP入口进行。在有网络的情况下, SUP把一部分功能分派给网络, 而把未被网络截获的部分转给本地系统。

SUP 也管理一个功能对其它功能的再调用。例如程序装卸(PL)和命令行解释 (CLI)功能要多次调用其它功能, PL和 CLI 同一般的用户程序没什么差别, 其中不包含一般用户程序不能执行的内容。把这类功能归入操作系统是因为它们具有公用性。而把它们划归SUP则有两个考虑, 一是使它们独立起来, 系统其它模块改变时不必修改它们。二是使之与未来的网络接口(也属SUP)处于同等地位, 即位于网络接口之外。因为网络要截取网络文件系统内所有对磁盘文件的访问, 而只将内存分配交由本地处理。

属于SUP的系统功能为12、47、50、59、107、150~155、163号功能, 请参看后面的系统功能汇总表。

任一进程在调用系统功能前须将功能号放入 CL 寄存器中, 并把地址和数据 分别 放到 DX、DS和DL寄存器, 如果地址和数据不需要由调用进程提供, 则只置功能号, 然后用一条 CALL指令使程序转到SUP的入口(相对段址0000H)

下面以CLI功能为例进一步说明SUP的职能。CLI的任务是解释并执行控制台命令。首先从命令行缓存器(CLBUF)取出命令, 接着为新创建的进程(或命令所涉及的 RSP)指定控制台。CLI接着调用 152 号功能对命令进行分析。如果命令前没有规定盘号, CLI首先查找同名队列, 一旦找到其名称与命令相符的队列, 随即打开该队列, 如果该队列是RSP型的, CLI查找与队列名相同的RSP, 并为其指定控制台。假如CLI未能找到RSP, 则转向磁盘上查找名为 CMD 的文件。如果没有规定盘号, CLI自动转向默认系统盘上查找, 但它只能打开系统态的文件。一旦成功地打开命令文件之后, CLI调用第59号功能装载程序到适当的内存空间。在装载成功并对基页(Bare Page)进行部分初始化之后, CLI继续分析命令后缀并将其填写到基页中。

CLI从内部的进程说明表中挑出一张空表, 连同用户数据区和96字节的堆栈均作初始化处理。之后再调用 144 号功能创建进程。CLI在上述各步调用中遇到任何差错将立即释放所占用的全部资源。

SUP位于系统的开始部分, 相对段地址为 0000H, 它既是SUP的入口, 也是所有功能调用的必经之路。为了与CPM-86兼容, MPM-86保留了直接调用BIOS的功能(功能号50)。因此在CPM-86下产生的调用BIOS功能的程序也能在MPM-86下运行, 但MPM-86只支持与字符设备接口的一类子程序, 而且需将子程序中的某些宗量如 Conin 和LIST按-XIOS的要求作适当转换。

三、实时监控(RTM)

RTM是控制MPM-86的核心, 它的任务主要是创建和终止进程、进程调度、进程通信、并处理逻辑中断系统以及I/O设备查询。

RTM模块包含的系统功能有0、131~145、156、157号功能。

1. 进程调度(Process Dispatching)

在一个多道操作系统中, CPU在任一给定的时刻均只能为一个进程服务。除非在程序中加以特殊处理, 否则必定存在各个进程对系统资源的竞争。操作系统处理这种竞争的方法是

挂起一个正在运行的进程而把资源让给另一个，RTM中有一个Dispatcher专门负责CPU的调度。在MPM-86下运行的每个进程均与两个数据结构相关联：进程说明书(PD)和用户数据区(UDA)。Dispatcher用这些数据结构保存和恢复进程的状态。每个进程在系统中可能的状态为：就绪，运行和挂起。就绪进程是那些只等待CPU的进程，而挂起的则是指那些等待系统其它资源或某一事件发生的进程。

一次调度过程可归纳如下：

- 1) Dispatcher将进程挂起，将其当前状态保存到PD和UDA中。
- 2) Dispatcher扫描就绪表(Ready List)，选出一个优先级最高的进程。
- 3) Dispatcher恢复所选进程的PD和UDA并将CPU给予该进程。
- 4) 进程执行到需要另一个资源或释放一个资源，或中断发生为止。这时又开始另一项调度过程以便让另一个进程运行。系统时钟每隔大约16毫秒(一次嘀嗒)产生一次中断，这也是CPU工作的切换间隔。

在调度时只有位于就绪表中的进程才够格候选。至于那些等待其它资源的进程在其要求未被满足前不可能运行。在MPM-86下，当一个进程处于下列任一等待情况下均将被阻塞：

- 等待队列消息以便完成一次读队列
- 等待队列的适当空间以便完成一次写队列
- 等待系统标志置位
- 等待一个可用的控制台或列表机
- 等待一段规定的时间以便脱离系统的延迟表(Delay List)
- 等待一个I/O事件结束。

Dispatcher在调度时把CPU给予优先级最高的进程，优先级同等时则采取轮流制。为了防止优先级高的计算型进程独占CPU资源，从而影响整个系统的性能，应该降低这类进程的级别。MPM-86要求至少有一个进程始终运行着，因此IDLE进程以很低的级别安放在就绪表上，当系统空闲时便执行这个空转进程。

进程说明书(PD)的格式如下：

相对地址 (HEX)	字节数	项目名	说 明
00	2	LINK	指示下一张说明书的地址，因此调用一次144号功能可以创建多个进程。
02	2	THREAD	确定说明书在PD表册中的位置，初始化时置0。
04	1	STAT	指明当前的进程状态 00 运行 01 查询 02 延迟 06 等待读队列 07 等待写队列 08 等待系统标志 09 等待连接字符I/O设备
05	1	PRIOR	当前优先级，为最高级。
			0-31 中断

相对地址	字 节	名 称	说 明
			32-63 系统进程
			150 初始化进程
			198 终端消息进程
			200 默认的用户优先级
			201-254 用户进程
			255 IDLE进程
06	2	FLAG	指明进程类型，按不同类型授予不同的特权。 详见[2]
08	8	NAME	进程名称，8个字节中的每一位都必须符合方可使用
10	2	UDA	进程的用户数据区地址
12	1	DISK	当前默认盘
13	1	USER	当前默认的用户号
1E	2	PARENT	创建本进程的父进程的PD地址
20	1	CNS	当前默认的控制台字符控制块
24	1	LIST	当前默认的列表机字符控制块

PD总共48个字节，以上未列出部分均留作内部使用。

进程本身运行时要用的数据放在256字节的用户数据区内，其中主要包括DMA地址，8086各寄存器的初值，堆栈地址，内存段地址以及中断向量等。其格式如下：

00h	RESERVED	DMA OFFSET	RESERVED
08h	RESERVED		
10h	RESERVED		
18h	RESERVED		
20h	AX	BX	CX DX
28h	DI	SI	BP RESERVED
30h	RESERVED		SP RESERVED
38h	INT 0		INT 1
40h	INT 2		INT 3
48h	INT 4		RESERVED
50h	CS	DS	ES SS
58h	INT 224		INT 225
60h	RESERVED		
68h	USER SYSTEM STACK		
⋮			
⋮			
F8h			

用户数据区

2. 队列管理(Queue Managment)

队列在 RTM 中分管进程间的通信, 同步和互斥。当一个进程要求调用驻留系统进程(RSP), 系统程序库(RPL)以及文件系统等公共资源时, 必须首先从相应的队列读取消息, 如果队列已空, 则表示资源被占用, 需等另一个进程把资源释放后方可调用, 一旦使用完毕立即把消息写回队列, 将资源释放。每个系统队列由队列说明书(QD)和队列缓存器(QB)组成, 缓存器用来存放若干固定长度的消息。队列与文件相似, 有建立, 打开读写和删除等步骤, 在MP M-86下的队列是一种以“存贮器文件”形式存在的特殊数据结构。

队列说明书格式

相对地址 (HEX)	项 名	字 节	
04	FLAGS	2	队列标志, 指明队列类型 0001 互斥队列, 一次只允许一个进程 0002 不可删除标志 0004 仅限系统进程使用 0008 RSP消息队列 0010 供内部使用 20 RPL地址队列 40 内部使用 80 内部使用
06	NAME	8	队列名称
0E	MSGLEN	2	队列消息的长度(字节数)
10	NMSGs	2	最多可容纳的消息条数
1A	BUFFER	2	队列缓存器地址。其长度必须等于NMS- GS * MSGLEN。若队列由 RSP 创建时, 应将 Buffer 地址填入此栏。由过渡进程 建队时, 由于其数据区不在SYSDAT以内, 故一般将此栏置为 0。

MP M-86 的每一个系统队列均有一张队列说明书位于系统数据区内。当调用134号功能建立队列时, 由调用者指明说明书的地址。此外系统数据区外面也有存放QD的队列数据表, 其大小由GENSYS决定。队列缓存区一般也必须在系统数据区内。队列名称是唯一的, 不允许重名队列存在。在对队列进行读写或删除前必须首先打开队列。打开队列(135号功能)的作用就是按照队列参量块(QPB)中设定的名字在现有的系统队列中查找名字相符的队列。如果找到则修改 QPB中的标识字段(QUEUEID), 指明队列已被打开。调用135号功能的进程必须给出QPB的地址。

QPB的格式如下:

00	保留	此栏和下一栏均由打开队列功能予以填写或修改
02	QUEUEID	
04	NMSGs	要读或写的消息数量
06	BUFFER	相对当前数据段的缓存区地址
08	NAME	8 字节队列名称

和QD给址的队列缓存区相对应,由QPB给址的寄存器在读队列时用来存放读出的消息,而在写队列时,则将内容送回队列缓存区。队列缓存区,位于SYSDAT区内,而后者则位于调用进程的当前数据区内。

MP M-86有两种读写队列的方式——无条件读写和条件读写,差别仅在于对调用不成功时的处理方式不同。无条件读某一队列时,如果队列是空的,则进程继续等待队列消息,而如果是条件读遇空队列时则返回一个错误信息。

当两个进程同时等待一个队列消息时,先满足优先级较高的进程,如级别相等则先来先得服务。

互斥队列是队列的一种特殊形式,它只包含一条消息。MP M-86中的互斥队列用大写的MX开头命名,例如MXdisk 是用来保证各进程依次访问文件系统的互斥队列。访问一个受互斥队列保护的资源的过程如下:

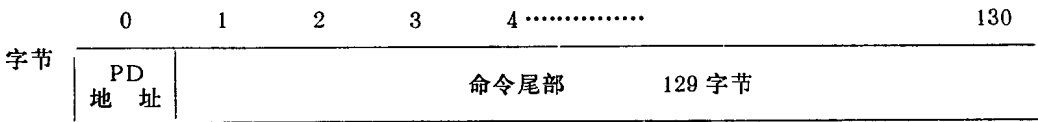
- 1) 进程调用无条件读队列功能,在得到队列消息前将自己挂起。
- 2) 收到消息后即访问资源。
- 3) 资源用毕后将消息写回队列,释放该资源。

如果进程在占有互斥队列时由于别的原因受阻。则RTM将自动将消息写回该队列,收回资源。

下面以RSP命令队列为例进一步讨论队列的实际功能及工作过程,作为这一小节的结束。

可以被控制台呼叫的或通过CLI(150号功能)调用的RSP称为命令RSP,必须为这种RSP建立一个特殊队列——RSP命令队列。这类RSP通常先执行某些初始化程序,包括建立和打开一个RSP命令队列,将进程的优先级改为过渡进程。然后进入一个循环程序(LOOP),LOOP的第一步是从RSP命令队列读消息。显然、写消息到队列的进程就是激活RSP的进程。一旦从调用读队列成功地返回之后,RSP便获得了所需要的资源,例如调用进程的控制台。当RSP完成了命令规定的任务后立即释放资源并重新进入读RSP命令队列的循环。命令RSP是个单一进程、只能被顺序地重复使用,因为队列中仅有一条消息,因此当几个进程同时呼叫一个单一进程时,它们将等待。

RSP命令队列的消息格式如下:



后 129 个字节是终端消息进程(TMP)发给CLI的内容(命令正名除外),头两个字节是调用进程的PD地址,RSP可利用这个地址获得有关调用进程的信息,例如默认的盘号,用户号以及I/O设备等。

当控制台接受命令之后,TMP执行一次CLI功能调用。由 CLI 打开一个与命令名相符的RSP型队列,并为PSP指定控制台,然后由CLI用命令尾部形成上述的RSP命令队列消息。过渡程序也可直接写消息到某个RSP命令队列,以同样的方式使用命令RSP。这样使用CLI功能的好处是先查找RSP,如果没找到,则只查找盘上的CMD文件。

3. RTM的其它功能

(1) 中断处理

当进程需要等待中断发生时, 它向一个逻辑中断设备发布一次 132 号功能调用(flag wait), 一旦所等的中断确实发生之后, 则通过调用133号功能(Set flag)“唤醒”进程。然后中断服务子程序转跳到Dispatcher, 由Dispatcher对所有在就绪表之外的进程(包括刚才被唤醒的进程)作重新调度。通常要处理中断事件的进程有较高的优先级, 因此能作出迅速反应。

(2) 设备查询

RTM中的 Dispatcher 也处理对非中断驱动型设备的查询, 在每次调度期间, 通过调用 POLL Device(131号功能)查询 XIOS 所等待的外部设备, 如前所述, 查询间隔大约为16mS, 虽然只需要查询那些进程要用的设备, 但实际上每次调度时都查询系统中的每台外设。

(参看XIOS一节)

四、存储器管理(MEM)

MEM掌管所有的存储器功能。MPM-86 2.0支持扩展的定长分片的内存管理, MPM-86以后的版本可能支持不同的管理方案。

MEM以片(Partition)为单位分配内存, 两个互不相关的程序不能在一个片内共存。如果所请求的程序段(Segment)大于规定的片长, 则由相邻的片合成一个连续的内存区。系统使用了最佳化算法, 这个算法使MEM在分配内存时能够在“片内”和“片外”碎片之间作出合理的折中。

如果片的长度划分得小, 会余下许多片外的碎片, 但却难以找到足以装下一个大程序的连续的内存区。反之如果把片长划得很大, 但要装的是个小程序, 结果在片内留下了碎片。虽然碎片可能足够装一个小程序, 但因为片号已被占用, 也不能被利用。MPM-86允许用户在系统生成期间, 根据实际硬件环境和使用要求, 合理地选择片的划分方案。对于以运行小程序而难得运行大程序的系统, 应当把片划分得小一些。而只要不是过分零碎仍可运行大程序。而对于主要运行大程序的系统, 则应把片划大。这种选择往往需要通过实验方能达到最佳。

系统内部用内存说明书(MD)来描述上述的存储片。初始化时把全部可用的存储片开列在一张名叫MFL的清单上。一旦被分配出去之后, 就把它从MFL上勾掉, 填到另一张MAL清单上。在MAL清单上既有以片为单位的记录, 也包含几片连在一起的连续内存区, 称作MAU, MAU由一片或若干片组成。MEM以MAU为单位进行管理。在接到一个内存请求时, 它首先发放MAU内的空片。但一个MAU不能给予两个互不相关的进程。假如一个进程被指定与另一个进程共用一个程序段, 则两者可共享一个含有那个程序段的MAU。MEM必须时刻记住某一个MAU归哪几个进程共享, 以确定这个MAU是否可以另行分配作它用。只有当MAU内的存储单元完全空闲时, MEM才把它所属的片送回MFL。

与MEM有关的系统功能为53~58和128~130号功能, 53~58从内部调用128~130、目的是为了与CPM-86兼容。

五、系统数据区(SYSDAT)的格式

相对地址 (HEX)	名称	字节数	用途说明
00	SUP入口	4	管理内部各模块间通信的 SUPERVISOR的入口地址, 所

			有内部系统调用均经此口。
28	XIOS入口	4	XIOS入口地址, 所有XIOS功能调用均经此口
2c	XIOS INIT	4	XIOS初始化入口地址, 通过调用对系统硬件进行初始化
38	PISPATCHER	4	DISPatcher的入口地址。
3C	PDISP	4	调度程序的入口地址, 执行一次JMPF PDISP等价于RETF指令。
40	MPMSEG	2	操作系统的首段地址
42	RSPSEG	2	RSP数据段索引表中第一个RSP的段地址。数据段的第一个字指向表中下一个RSP。系统一经初始化之后, 此栏为0。
44	ENDSEG	2	操作系统以外的首段地址
47	NCNSVED	1	由GENSYS设定的控制台数目
48	NLIST	1	由GENSYS设定的列表机数目
49	NCCB	1	由GENSYS设定的字符控制块的数目
4A	NFLAGS	1	由GENSYS设定的系统标志数
4B	SYSDISK	1	默认的系统盘
4C	MMP	2	每个进程能占用的最大内存区
4E	NSLAVE	1	网络数
4F	DAYFILE	1	=FFH 显示登录日期, 由GENSYS选择
50	TEMP DISK	1	默认的临时盘用于建立临时文件
51	TICKS/SEC	1	每秒的系统滴嗒数(时间片数)
52	LUL	2	BDOS上锁清单(LOCK LIST)的地址
54	CCB	2	字符控制块(CCB)表格地址
56	FLAGS	2	标志表地址
58	MDUL	2	冗余的存储器说明书表册首址
5A	MFL	2	空闲的存储器表册首址
5C	PUL	2	冗余的进程说明书表册首址
5E	QUL	2	冗余的队列说明书表册首址
60	QMAU	8	队列缓存器地址分配
68	PLR	2	待运行的进程说明书(PD)表册地址
6A	DAR	2	需延迟一定时间的PD表册地址
6C	DRL	2	刚就绪的PD缓存区
6E	PLR	2	正在查询设备的PD表册地址

72	THRDRT	2	系统当前所有进程总表册地址,清单通过THRDRT装订成册。
7C	QLR	2	系统所有队列说明书(QD)表册地址
7E	MAL	2	可用的内存分配表
78	VERSION	2	字符串地址
7A	VERNUM	2	MPM-86版本号 (功能12)
7C	MPMVERNUM	2	MPM-86版本号 (功能163)
7E	TOD DAY	2	天数
80	TOD HR	1	小时
87	TOD MIN	1	分
82	TOD SEC	1	秒
83	NCON	1	控制台数
84	NLST	1	列表机数
85	NC10	1	NCON+NLIST

六、扩展BIOS(XIOS)

DR公司发表的MPM-86—2.0的硬件背景是Intel的SBC89/12单板机, Intel1204磁盘控制器和一个Intel SBC534通信扩展板。与硬件直接有关的子程序全部集中在XIOS模块中。这些子程序可被修改, 以使整个操作系统适应各种8086或8088型磁盘系统。

XIOS有两个入口INIT和ENTRY, 分别位于XIOS代码段的0H和03H。

INIT是初始化程序的入口, 在系统初始化期间由INIT进程调用。MPMLDR把MPM·SYS装到MPM·SYS文件头指明的内存区。初始化自动把这个地址放到CS和DS寄存器中, (在CP/M-86下用DDT装入时必须键入这个地址)。MPMLDR由此转跳到0地址, 开始执行MP/M-86的SUP模块的初始化程序, 初始化步骤如下:

1) 创建INIT和IDLE进程, 以下的工作由INIT进程执行, 其优先级等于1。

2) INIT进程调用系统各模块的初始化程序, 每个模块的第一行代码是转跳到自身初始化的指令、依次进行、XIOS的初始化排在最后。初始化程序对硬件和中断向量进行初始化和设置初始值。

3) 从XIOS返回SUP, SUP初始化程序(运行在INIT进程)创建某些队列, 调用MXCONSOLE和MAXLIST, 启动RSP。INIT进程结束初始化之后, 所有对XIOS的访问均通过Entry入口。

MPM-86的XIOS的功能主要分为字符I/O和磁盘I/O两大类制。

1. 字符I/O功能 这部分主要包括功能1、2、5、11、13、15、17、和20、21。

当一个在MPM-86下运行的进程需要同字符I/O设备(控制台或列表机)交换信息时, 由于速度不匹配, 迫使进程处于等待状态。在单用户的CPM-86下, 用一个让CPU空转的LOOP来等待I/O设备把数据准备好。但在多用户系统中显然不能这样浪费系统资源。MPM-86用两种方法等待I/O设备就绪, 对于靠中断驱动的外设使用FLAGWAIT和FLAGSET两个功能调用。当进程等待中断发生(I/O设备号由中断向量识别)时, 调用FLAGWAIT将等待标志置位, 然后让出CPU给别的进程, 当中断发生时, 由中断子程序调用FLAG

SET, 将中断标志置位, 调度程序根据这个标志信息随即把 CPU 分配给等待中断的进程。FLAGSET 是中断子程序可调用的唯一系统功能。系统有一个标志表, 每个进程有自己对应的标志号。第 2 种方法是通过设备查询(POLL DEVICE)。申请 I/O 设备的进程调用 POLL DEVICE 功能, 如果设备尚未准备好发送或接受一个字符, 则将该查询进程置于查询表(POLL LIST)中, 系统在每个调度间隔调用 POLL DEVICE, 并修改查询表, 一旦条件满足, 便将进程从查询表移到就绪表中, 等待 CPU。每台设备有一个设备查询号, 系统按号进行查询和唤醒进程。

在 MPM-86 中有一个独立的字符 I/O 模块(CIO)专门处理所有的控制台和列表机的输入输出。而 XIOS 中所包含的仅是最下层的字符 I/O 处理功能。CIO 为每台 I/O 设备准备了一个描述该设备的字符控制块(CCB)。CCB 中包含下列信息: 当前属主、等待该设备的进程表(进程说明书表册的首址), 行编辑变量以及状态信息。CCB 位于系统数据区的 CCB 表格区。每一张进程说明书都包含其默认设备的 CCB 地址。CCB 的编号顺序为 0 号。控制台对应 0 号 CCB, 列表机的 CCB 排在控制台之后。CCB 的数目由系统生成时选定。

2. 磁盘 I/O 功能

这部分的主要功能列表如下:

名 称	功 能
HOME	磁头回到零磁道, 如磁盘控制器不支持这一功能, 可改用 SETTRK
SELECT DISK	选择当前驱动器, 返回时给出所选驱动器的盘片参量信息头(DPH)的地址, 若使用标准 8 吋软盘则无须改变信息头及其有关表格的内容。选盘的实际物理操作在完成读写操作后进行
SETTRACK	设置磁道号, 在下次读写操作后开始查找磁道号。标准软盘的选择范围为 0—76, 非标准磁盘子系统为 0—65535
SETSECTOR	设置扇区号
SET DMA OFFSET	DMA 地址偏移量, 与基地址合起来指向一个 128 字节的缓存区
SET DMA BASE	DMA 缓存器基地址
READ	从当前盘读一个扇区的数据到缓存区, 如返回错误信息, 则继续试读若干次, 能否恢复取决于错误类型。
WRITE	把缓存区的数据写到指定的盘区
SECTOR TRNSLATE	逻辑扇区号转换成物理扇区号, MPM-86 是以标准的 IBM37408 软盘形式出售的, 它的 SKEW 因子是 6。两次接续的读写之间实际相隔 5 个扇区。物理扇区号通过查表获得。
FLUSH BUFFERS	请 XIOS 内部的 BLOCKing 缓存区

系统通过调用上述XIOS的功能实现对磁盘的访问。在执行对磁盘的读写操作之前,通过一系列调用以确定盘号,道号扇区号以及DMA地址。选择驱动器一般只在要改变驱动器前调用一次。DMA的地址包括数据段基地址和偏移量两部分。只有当改变DMA地址时才调用这两个功能。而磁道和扇区则必须在每次读写前选择。读写功能在返回错误信息前自动重复若干次(标准是10次)。

3. 描述磁盘特性的参量表

每个盘驱动器通常都有一个16字节的磁盘参量头(DISK Parameter Header),给出有关盘驱动器的信息并为某些BDOS操作提供工作单元。DPH的格式如下:(每项均为2个字节)

XLT	逻辑扇区号与物理扇区号对照表地址
0000	供BDOS使用的工作寄存器
0000	供BDOS使用的工作寄存器
0000	供BDOS使用的工作寄存器
DIRBUF	128字节缓存区地址,供BDOS处理目录项时使用,所有DPH公用一个缓存区
DPB	盘驱动器参量块(DPB)地址,盘片特性相同的驱动器使用同一个DPB
CSV	BDOS缓冲区地址,用于软件判查盘片是否改变,每个DPH有一个独立的缓冲区。
ALV	BDOS缓冲区地址,用于保存盘片存贮的地址信息,每个DPH有不同的地址。

每个DPH描述一个盘驱动器,这样对于N个盘驱动器,构成一张DPH表,格式如下:

DPBASE

00	XLT 00	0000	0000	0000	DIRBUF	DPB 00	CSV 00	ALV 00
01	XLT 01	0000	0000	0000	DIRBUF	DPB 01	CSV 01	ALV 01

(and so-forth through)

n-1	XL Tn-1	0000	0000	0000	DIRBUF	DPBn-1	CSVn-1	ALVn-1
-----	---------	------	------	------	--------	--------	--------	--------

DPBSE是相对于操作系统首址的地址偏移量。调用选择驱动器(SELDSK)功能时,返回的地址偏移量就指向一个所要选的驱动器的DPH。XLT00=XLTn-1给址的扇区地址对照表位于XIOS的另一处,每张表按逻辑扇区号排列,一一对应。

盘片参量块(DPB)比DPH复杂,每个DPB可由一个或几个DPH寻址。其格式如下:

	字节	逻辑
SPT	2	每道逻辑扇区数,每128字节为一个逻辑扇区
BSH	1	BSH和BLM均为描述盘块大小的因子,若盘块包含8个扇区,则BSH=Log _e 8=3, BLM=8-7=1
BLM	1	

EXM	1	FCB中的范围号换算因子,其值等于每个FCB能容纳的最大范围数减1。通常每个FCB最多能控制16个盘块,若盘块为1K字节,则恰好为1个范围号,如果盘块为2K、4K、8K字节时,每个FCB所容纳的范围数便相应增加。
DSM	2	每个盘片包含的最大盘块号。因为第一个盘块为0号,故DSM等于盘块总数减1。
DRM	2	最大目录项号,等于目录项总数减1。每个目录项占32字节。
ALO、1	2	目录区盘块号地址映像。2个字节中的每一位对应一个目录盘块。1表示可用。
CKS	2	目录项占据的扇区数,对于软盘 $CKS = (DRM + 1) / 4$,用于在系统热启动时检验盘片是否改变,对于硬盘,此项为零。
OFF	2	磁道偏移量,物理磁道号等于偏移量加BDOS请求的逻辑道号。OFF可用于把一个盘片划分成几个逻辑盘区。

一个8"软盘的DPB列于下表

SPT	26
BSH	3
BLM	7
EXM	0
DSM	2.42
DRM	63
ALO	11000000
CKS	16
OFF	2

上述数据表中的值可以利用一个叫GENDEF的程序自动产生,但为了分析核对,了解每一栏的含义及其相互关系还是必要的。

BSH和BLM两项决定了盘块大小(BLS)

BLS	BSH	BLM
1K	3	7
2K	4	15
4K	5	31
8K	6	63
16K	7	127

EXM的值取决于DSM值大于或小于256

BLS	EXM (DSM < 256)	EXM (DSM > 256)
1K	0	
2K	1	0
4K	3	1
8K	7	3
16K	15	7

DSM是给定驱动器所能支持的最大盘块数，以BLS为单位。总字节数为 $BLS * (DSM + 1)$ ，这个值必须限制在实际盘片的总容量以内，不包括为操作系统保留的磁道。DRM 是目录项总数减1，它决定了ALO和ALI的值。ALO和ALI合起来形成16位的字位图，每一位代表一个可用的目录盘块。每个目录项占32字节，如果 $DRM = 127$ (有128个目录项)， $BLS = 1K$ ，则每个盘块包含32个目录项，总共需保留4个盘块存放128个目录项。

DPB是描述磁盘特性的数据结构，如果磁盘特性相同，则可使用同一个DPB，因此同一个DPB可被若干DPH寻址。当需要使用新驱动器时，只需改变DPH中的地址指针，BDOS的SELDSK功能将DPB值复制到指定的内存区。

在DPH中有两个地址值CSV和ALV，分别指向XIOS后面的未初始化的内存缓冲区，它们的长度由DPB中的CKS和DSM决定。前一个用来保存目录检验信息(Checksum)，如果 $CKS = (DRM + 1) / 4$ 则 CSV 所寻址的缓冲区必须等于 $(DRM + 1) / 4$ 个字节。根据前面的定义， $(DRM + 1) / 4$ 是目录项占据的扇区总数。后一个缓冲区长度为 $(DSM + 1) / 8$ ，等于盘片的总盘块数。BDOS用它来保存盘束分配信息。

七、MPM-86的文件系统

基本磁盘操作系统(BDOS)能管理1~16个逻辑盘上的有名文件。每个逻辑盘由目录区和数据区构成。目录区在逻辑上又划分为16个区，分别对应于16个用户号(0~15)。通常系统只显示当前用户号下的文件。BDOS为文件分配空间或收回已不用的盘区都是自动进行的。内部盘区的组织与分配情况对调用进程是透明的。

BDOS的功能主要有以下四类：

- * 文件存取 建立新文件，打开和关闭一个现有的文件以顺序方式或随机方式读或写文件，一般以128字节为基本读写单位。

- * 目录处理 目录查找、文件删除、文件改名、设定文件属性、为文件规定一个口令以及计算一个文件的大小。只有查找和删除时可以使用文件的泛指符号(* 或?)，BDOS不允许对一个已经打开的文件进行删除、改名和设定属性。

- * 驱动器控制 选择系统默认的驱动器、计算驱动器的自由空间，查询驱动器的状态，设定驱动器的目录标记以决定文件是否需要口令，日期和时间的刻印方式以及复位驱动器。

- * 其它功能 设定当前的DMA地址，访问和修改当前用户号，挂接到一个新程序，请内部读写缓冲区，规定每次读写的扇区数和设定BDOS的出错处理方式。BDOS每次可以读写1—16个记录、隐含值为1。BDOS可以截获错误自行处理，也可以把出错信息全部返回给调用进程。

MPM-86的文件命名规则大体上同CPM系统一样。下面列出常用的文件类型