

INTEL iAPX86,88
iRMX86 操作系统

李宝山 编译

一九八三年二月

前 言

操作系统是计算机软件的重要组成部分，它是用软件手段增加计算机功能的一种工具。操作系统为提高计算机利用率、方便用户、提高计算机的响应时间提供了有利条件。用户通过操作系统去使用计算机，因此，操作系统又是用户和计算机之间的桥梁与接口。操作系统的主要功能是管理中央处理机、存贮器、外围设备和信息，控制计算机作业的运行及处理中断等。另外，它也控制编译程序、解释程序、编辑程序、连接装配程序等系统软件和各种应用程序的运行。

最早的操作系统是为IBM704计算机配置的FORTRAN监督程序系统。从前的计算机速度比较慢，使用范围较小，大都是单用户独占计算机全部资源。随着计算机技术的飞速发展，现在的计算机与第一代计算机速度相比，已经提高一万多倍。其应用范围也深入到社会生活的各个方面。如果仍用过去的方式操作，让计算机等待人工操作，或者输入输出，就会使硬件速度的提高失去意义。为解决这一矛盾，各种各样的操作系统应运而生。

微型计算机操作系统出现于70年代中期。1971年世界上产生了第一台微处理器—INTEL4004之后，1978年又发表了INTEL8080微处理器，时隔一年左右MOTOROLA公司研制出M6800，1975年ROCKWELL公司发表R6500，1976年ZILOG公司推出Z—80微处理器。1978年以后又相继出现了INTEL8086、M68000、Z—8000为代表的第三代十六位微处理机。目前，32位微处理机也已广泛使用。微处理机的飞速发展迫切要求为微型计算机系统配置操作系统。1976年DIGITAL RESEARCH软件公司研制了较早的微型机操作系统CP/M (Control Program/Monitor)，这个操作系统短小，简练，适应性强，但功能不够完善。CP/M操作系统出现以后，各式各样的微型机操作系统纷纷问世。微型机操作系统就其发展过程来说，大体分为两个阶段。第一阶段是1976年—1979年，这一时期以发展单用户、单道作业的操作系统为主。第二阶段是1979年—1982年为多用户、多作业、分时操作系统发展阶段。目前，微型计算机操作系统正日臻完善。

本书介绍的是美国INTEL公司1982年发行的iRMX86实时多任务操作系统 (Real-time Multi-tasking EXcecutive)。这个操作系统是为INTEL iSBC86,88系列单板计算机和其它以iAPX86,88为基础的计算机而设计的软件包。它功能齐全、通用性强、使用方便，是比较好的一种微型机操作系统。全书共分五章。第一章介绍iRMX86操作系统的概况及其各个组成子系统。第二章介绍iRMX86操作系统运行时所需要的硬件环境。第三章通过举例简单说明该操作系统的应用。第四章较详细地叙述iRMX86操作系统的命令格式及其功能。第五章介绍iRMX86操作系统的通用开发接口UDI的各个系统调用格式。书末还附有三个可供查阅的附录。

承蒙杨廷善同志对本书做了详细审阅，对其中不妥之处给予了指正。在此表示衷心地感谢。

由于编者水平有限，加之时间仓促，一定会有许多错误和不妥之处，请读者批评指正。

李宝山 1983年2月

目 录

第一章 系统概述	(1)
1—1 iRMX86操作系统结构.....	(1)
1—2 iRMX86系统的硬件环境.....	(9)
1—3 语言翻译程序和应用程序.....	(11)
1—4 通用开发接口 (UDI)	(11)
1—5 iRMX86文件系统.....	(11)
1—6 程序加载.....	(14)
1—7 引导加载.....	(15)
1—8 监督程序.....	(15)
1—9 检错处理.....	(15)
第二章 硬件准备	(16)
2—1 iRMX86硬件环境.....	(16)
2—2 板的修改.....	(17)
第三章 iRMX86系统的应用	(25)
3—1 启动系统.....	(25)
3—2 使用iRMX86命令.....	(25)
3—3 前置位参数.....	(26)
3—4 终端控制.....	(27)
3—5 在输入表和输出表中文件数不相等时系统的处理规定.....	(27)
3—6 iRMX86命令举例.....	(28)
第四章 iRMX86命令.....	(34)
4—1 命令语法格式.....	(34)
4—2 iRMX86命令格式.....	(36)
1. ATTACHDEVICE.....	(36)
2. BACKUP.....	(37)
3. COPY.....	(38)
4. CREATEDIR.....	(40)
5. DATE.....	(41)
6. DEBUG.....	(41)
7. DELETE.....	(42)
8. DETACHDEVICE.....	(43)
9. DIR.....	(43)
10. DISKVERIFY.....	(49)
11. DOWNCOPY.....	(50)

12.	FORMAT.....	(51)
13.	RENAME.....	(53)
14.	RESTORE.....	(54)
15.	SUBMIT.....	(55)
16.	TIME.....	(57)
17.	UPCOPY.....	(57)
	第五章 UDI系统调用	(59)
5-1	UDI的使用.....	(59)
5-2	系统调用.....	(60)
1.	DO \$ ALLOCATE.....	(63)
2.	DO \$ ATTACH.....	(64)
3.	DO \$ CHANGE \$ EXTENSION.....	(64)
4.	DO \$ CIOSE.....	(64)
5.	DO \$ CREATE.....	(65)
6.	DO \$ DECODE \$ EXCEPTION.....	(65)
7.	DO \$ DELETE.....	(65)
8.	DO \$ DETACH.....	(66)
9.	DO \$ EXIT.....	(66)
10.	DO \$ FREE.....	(66)
11.	DO \$ GET \$ ARGUMENT.....	(66)
12.	DO \$ GET \$ CONNECTION \$ STATUS.....	(67)
13.	DO \$ GET \$ EXCEPTION \$ HANDLER.....	(68)
14.	DO \$ GET \$ SIZE.....	(68)
15.	DO \$ GET \$ SYSTEM \$ ID.....	(96)
16.	DO \$ GET \$ TIME.....	(69)
17.	DO \$ OPEN.....	(69)
18.	DO \$ OVERLAY.....	(70)
19.	DO \$ READ.....	(71)
20.	DO \$ RENAME.....	(72)
21.	DO \$ SEEK.....	(72)
22.	DO \$ SPECIAL.....	(73)
23.	DO \$ SWITCH \$ BUFFER.....	(73)
24.	DO \$ TRAP \$ EXCEPTION.....	(74)
25.	DO \$ TRUNCATE.....	(74)
26.	DO \$ WRITE.....	(75)
	附录A: iRMX86状态码	(76)
	附录B: iRMX86系统调用	(80)
	附录C: 加载程序和监控程序命令表	(84)

第一章 系统概述

iRMX86操作系统是管理和扩展iSBC86单板计算机资源的软件包，是美国INTEL公司的实时多任务操作系统。它操作在INTEL iSBC86/12单板计算机和iAPX86(8086)为基础的用户单板计算机上。iRMX86PC操作系统是iRMX86操作系统的一个通用板本。它是专门设计用于开发和运行程序的。它为iAPX86(8086)用户提供了一个简单易用的工具，为硬件系统的应用提供了一个广阔的天地。

1-1 iRMX86操作系统结构

iRMX86操作系统是可组合结构型操作系统。除核心部分外，系统的其他部分可根据用户实际应用时的需要而决定其取舍。iRMX86操作系统结构如图1-1所示。它由许多层次构成。下面简要说明各层次名称及功能。

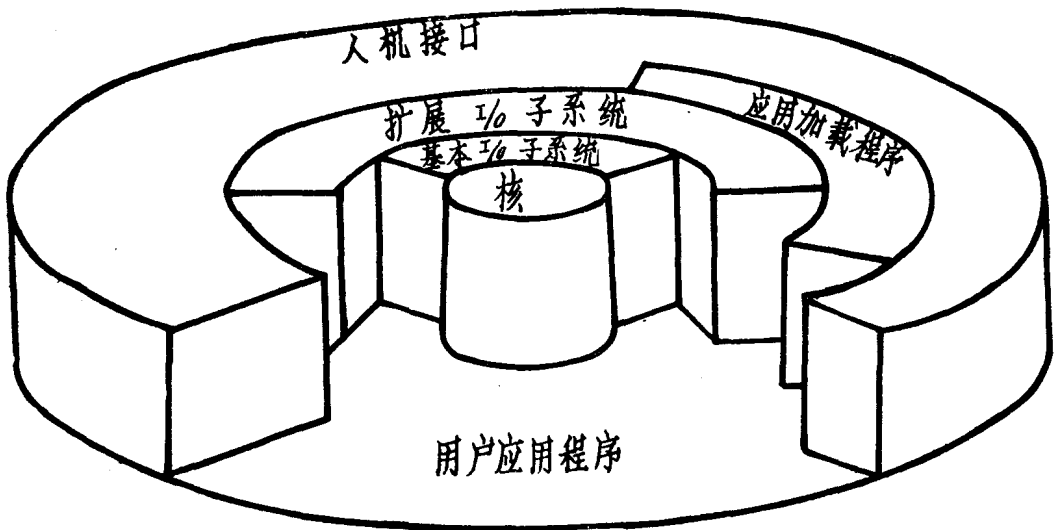


图1-1 iRMX86 操作系统层次图

1. 核:

核是iRMX86操作系统的核心。它提供建立实际应用的iRMX86操作系统的功能，也是每一个实用iRMX86操作系统必不可少的组成部分。它主要完成处理器管理和编制目录、中断管理、内存管理、目标控制以及错误管理等各种功能。为完成这些管理和控制功能，共有三十多条核系统调用指令可供使用。

核提供建筑块，由这些建筑块可以构成其它的子系统（如基本I/O系统、扩展I/O系统、

应用加载程序及人机接口等)和应用系统。这些建筑块称为目标,目标分为下列各种类型:

- 任务
- 作业
- 段
- 邮箱
- 信号
- 区域
- 扩展目标
- 组合目标

下面分别说明一下各种类型的意义:

① 任务是系统中的现役目标。它们执行系统的工作。
② 作业是任务从事工作的环境。这个环境由任务、任务所用的目标、任务能编目的目录及内存区组成。

- ③ 段是一段内存区,是任务用以通讯和存放数据的媒介。
- ④ 邮箱供任务存取另外的目标使用。
- ⑤ 信号区可使一个任务向另外的任务发送信号。
- ⑥ 区域用以防护某些指定的分享数据。
- ⑦ 扩展目标用以设置新的目标类型。
- ⑧ 组合目标是由扩展目标设计的新类型的目标。

最后三个目标类型是为系统程序员备用的。

核提供的系统调用是一种操作符,任务用这种操作符来操作目标。

当使用系统调用时,任务提供参数值。例如标记、名字或其它值,这决定于系统调用应用时的需要。任务可用系统调用形成如下功能:

- 建立目标。
- 删除目标。
- 送信息给其它任务。
- 接收其它任务的信息。
- 得到关于目标的信息。
- 用名字编目目标。
- 从目录中删除目标。

目标有五种类型,它们都具有统一的特性。这五种目标类型是:

① 任务:它有两种功能,主要功能是去做指定的功能。其次是得到处理机的独享控制权,使其主要功能向前推进。

核的主要功能之一是裁决几个等候处理机独享控制的任务的竞争结果。核通过记忆每一个任务的执行状态和优先级来完成这个功能。在某一时刻,每一个任务的执行状态可以是运行、准备、休眠、挂起或休眠挂起状态之一。每一个任务的优先级是一个整数值。这个值在0—255之间,0是最高优先级。核实行的裁决法则是,具有最高优先权的准备任务是运行任务。从核这方面考虑,一个任务可以看作是纯粹的数值顺序。其中包括:

- 任务优先权。

- 任务执行状态。
- 包含该任务的作业的标记。

当一个任务成为运行任务时，依次发生如下操作：

- 先前运行的任务由核按序保存。
- 核设置新的任务运行顺序。
- 新的任务开始执行。

任务继续运行，一直运行到下列事件发生时才停止：

- 任务退出准备状态，如挂起或删除。
- 更高优先权任务进行准备状态，先前的运行任务退出运行状态。

② 作业：作业由任务和任务需要的资源组成。

在系统中，作业形成一棵族树，除了根作业，树中的每一个作业都从其母体得到资源。在用户作业中的任务可以建立附加的作业。如果建立了附加的作业，就会使作业树变得更大。

作业树的最初形状如图1—2所示

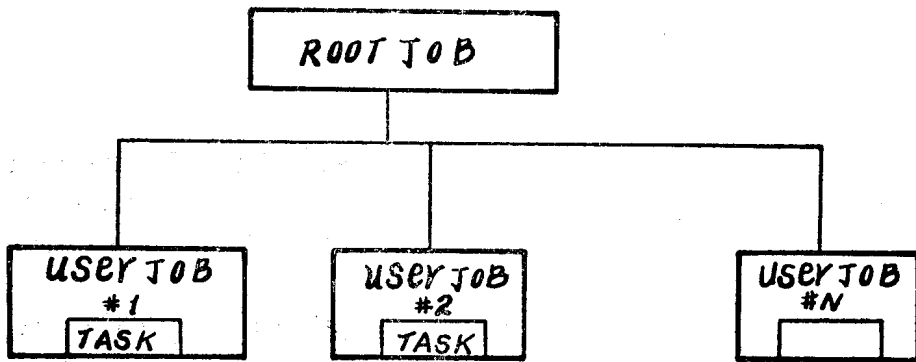


图 1-2

作业树图形

同每一个作业相关的是一个目标目录。对一个作业来说，目标目录是一段内存空间，在这段内存空间中，任务在一个名字下面编一个目标。此后，认识这个名字的其它任务就可以使用这个目录对该目标进行存取。

同每一个作业相关的还有内存源(memory Pool)。这是定位作业和它的下属的一定数量的内存空间。在作业中建立目标所需要的所有内存都从该内存源中来。

③ 段：任务需要的基本资源是内存。内存以段的形式分配给任务。需要内存的任务申请它所需要的大小的段。当作业建立时，核从内存划出一段提供给该任务的作业。

④ 邮箱：邮箱是可以用于交互任务通讯的两类目标之一。当任务A送一个目标到任务B

时，任务A必须送这个目标到邮箱中，而任务B则访问邮箱。如果目标不在邮箱内，它就再等待要求的一段时间。用这种管理方法送一个目标能达到各种要求。目标可以是一个包含正在等待的任务所需要的数据的段。另一方面，段可以是空白，可以由它构成一个信号送给等待任务。目标还可以指出接收任务的目标。

⑤ 信号：信号是一个抽象的基元管理员。它分配基元给申请它的任务，并接收由任务来的基元。信号的典型例子是相互拒斥。假定应用系统包含一个I/O设备，它用于多个任务输出。为保证在某一时刻只有一个任务使用该设备，用户可以按装信号基元，在使用设备之前，让任务得到这个基元。一个等待使用设备的任务，将从信号中申请基元，当它得到基元时，它就使用设备，然后将基元送回信号，当任务使用设备时，信号没有基元，这就有效地排斥了其它任务对设备使用的可能性。

核除了对目标进行管理的功能外，还具有对两种特殊事件进行管理的功能。这两种特殊事件是故障状态和中断。

在iRMX86操作系统进行系统调用时，其结果不一定是予想的结果。例如，内存不能满足任务申请或使用了无效参数等。这就引起故障状态。故障状态发生后，控制被传送到与当前任务相关的故障管理程序。故障管理程序是一个过程，这个过程通过下列方法解决问题：

- 校正发生的问题并试图再试一次。
- 仅仅记录错误。
- 删除或挂起出错的任务。

iRMX86操作系统可以提供两级综合的错误管理：等级错误管理和选择性错误处理。

等级错误管理允许任务管理在系统不同级别发生的错误。每一个应用程序可以通过错误管理程序选择处理错误的方法。错误可以分为两类：环境状态错误（如内存空间不足）和程序员错误（如使用了无效参数等）。

为实现实时系统功能，iRMX86操作系统提供两级中断管理：中断管理程序和中断任务。中断管理程序响应时间较好，而中断任务响应能力较强。中断任务允许使用iRMX86的所有系统调用并只屏蔽较低优先权中断。中断管理程序允许直接控制中断且只允许使用中断系统调用。如果系统需要多于8级的中断时，iRMX86操作系统允许增加到7个从属的8259A中断控制器，此时系统可实时响应多达57级中断源。

2. 基本I/O子系统。

基本I/O子系统是一个可选择性子系统。

iRMX86操作系统提供两种可供选择的I/O系统。其中之一是基本I/O子系统，另一种是扩展I/O子系统。

大多数的应用都要同外设如行式打印机、终端、软盘驱动器、磁泡存储器或遥控设备等进行通讯，但每一应用系统不可能都有同样的外设。为适于不同用户的不同要求，iRMX86操作系统提供两种I/O子系统供用户选择。

基本I/O子系统是两种I/O系统中适应性更强的一种子系统。它提供很强的适应能力，其中包括：

- 允许用户设计和采用自己的缓冲技术。
- 对设备驱动程序提供异步接口，允许用户用其它操作显式选合I/O功能。

· 提供用户任务控制细节：基本I/O子系统的系统调用常常有许多参数。使用这些参数，用户任务可以变化每一个系统调用的功能以满足用户应用系统要求的不同特性。

3. 扩展I/O子系统。

扩展I/O子系统提供比基本I/O子系统更高级的接口且更容易使用。它具有如下特征：

· 自动缓冲式I/O操作：扩展I/O子系统对所有I/O提供自动缓冲操作方式。除了指定扩展I/O子系统使用的缓冲器数目外，用户任务不必用缓冲方式调用。如果用户应用系统不需要缓冲，用户任务可以通知扩展I/O子系统不使用缓冲器。

· 同步系统调用：扩展I/O子系统提供同步系统调用，减小了用户应用系统的复杂性，有助于降低开发费用。虽然扩展I/O子系统调用是同步方式，用户仍然可以使用迭合式I/O操作。用户任务只需通知扩展I/O子系统使用缓冲器，扩展I/O将自动地用迭合式I/O操作。扩展I/O子系统的系统调用需要很少几个参数，这就使用户任务从冗长的细节中解放出来，并减少了成本。

4. 应用加载程序。

应用加载程序是从第二存储器加载代码到RAM中去的一种有效工具。它的主要特性是：

- 设备独立性。
- 加载三种代码的能力：绝对代码、位置独立代码、加载时可定位代码。
- 同步、异步系统调用。
- 支持程序迭合写操作。
- 可结构性。

——绝对目标代码是可能出现的三种代码之一。一个绝对目标码模块是经LOC86处理过的只能在内存的一定位置运行的代码。应用加载程序可将绝对目标码模块加载到指定的内存位置。

——位置独立代码（通常称为PIC）不同于绝对目标码，它可以被加载到内存的任何位置。当应用加载程序被申请加载PIC时，加载程序得到iRMX86段并加载PIC到该段中去。PIC的优点是不需要为某个模块保留固定的内存空间。

——加载时可定位代码是第三种代码格式。通常称为LTL。LTL类似于PIC，它可以被加载到内存的任何地方。不同的是它可以用于几个代码段或几个数据的任务。

应用加载程序可从任何支持iRMX86命名文件的设备中加载目标代码。用户iRMX86操作系统可由在下列设备上支持命名文件的设备驱动程序来引渡：

- iSBC204 单密度软盘控制器。
- iSBC206 硬盘控制器。
- iSBC208 单、双密度软盘控制器。
- iSBC215 温氏盘控制器。
- iSBC218 软盘控制器。
- iSBC220 SMD硬盘控制器。
- iSBC254 磁泡存储器板。

如果用户需要从一个INTEL未提供设备驱动程序的设备加载，用户可以增写自己的设备驱动程序。

5. 引导加载程序。

引导加载程序的目的是提供一种方法。当iAPX86处理机被复位时，可以用这种方法从第二存储器中加载一部分或所有用户应用系统到RAM中去。利用这种方法，可以把应用软件和大量iRMX86软件存放在第二存储器中而不是必须存放在ROM中。无论何时，只要处理机复位，引导加载程序就加载操作系统和应用软件到RAM中，并将控制传递给操作系统。

引导加载程序由两部分组成——第一阶段程序和第二阶段程序。只有第一阶段程序驻留在ROM中。当iAPX86微处理机复位时，它首先开始运行。第一阶段程序的目的有如下意义：

第一，它确定哪一个第二存储器被加载。

第二，它确定哪一个文件被加载。

第三，它加载并传递控制给第二阶段程序。

第二阶段程序存放在第二存储器中，它驻留在引导加载程序从其上加载用户软件的设备上。第二阶段程序通过下列步骤完成整个引导过程：

第一，它将自身读到主内存中。

第二，它找到被加载的文件（该文件包含操作系统和应用软件）。

第三，它将文件加载到主内存中。

第四，它将控制传递给加载的文件。

引导加载程序可以由任何种类的第二存储器设备使用。如硬盘、软盘、磁泡存储器、磁带等。但对任何一种用户希望使用的引导加载程序的设备来说，用户必须有一个设备驱动程序。设备驱动程序是过程的集合，它允许引导加载程序同包含被加载文件的设备进行通讯。引导加载程序所需要的设备驱动程序不同于基本I/O和扩展I/O子系统所需要的驱动程序。iRMX86包含下列随机存取设备的设备驱动程序：

- iSBC204 软盘控制器。
- iSBC206 硬盘控制器。
- iSBC208 软盘控制器。
- iSBC215 温氏盘控制器。
- iSBC218 软盘控制器。
- iSBC254 磁泡存储器控制器。

由于这些驱动程序是iRMX86产品的一部分，用户仅仅按装这些驱动程序到引导加载程序中即可。

6. 人机接口。

用户通过在终端打入命令和接收信息而同应用设备进行联系。iRMX86操作系统提供大量的这种命令和信息。这些命令信息称为人机接口。

人机接口是iRMX86操作系统的最外一层。它为用户提供大量应用程序，也为应用程序员提供大量产生用户应用程序的手段。

iRMX86操作系统提供的人机接口命令可形成如下功能：

- 建立目录文件。
- 建立拷贝。
- 删除。
- 重新命名文件。
- 加载和启动应用程序。

- 格式化设备卷。
- 以批模式执行命令文件。

人机接口还提供一些在调试中有用的应用程序。如能使用户借助于系统调试程序而启动命令的调试命令，用于同iSBC957A/B监控程序联系起来以便转换MDS文件为iRMX86文件格式或转换iRMX86文件为MDS文件格式的拷贝手段。

使用人机接口，用户可以从终端通过说明一个控制程序的文件名来调用一个程序，同时，人机接口也提供大量的系统调用。

7. 调试程序

几乎每一个应用软件的开发都需要调试，为开发iRMX86应用系统，INTEL提供了三种调试工具：iRMX86调试程序、ICE-86在线仿真器和iSBC957A/B监控程序。

调试程序有它自己的终端管理程序；用户应用软件可以使用调试程序的终端管理程序。

调试程序有如下功能：

- 调试程序可用作任务，作业或系统故障管理程序。
- 观察iRMX86目标表。这个表包括作业、任务、准备任务、挂起任务、休眠任务、任务队列的改变、邮箱目标队列以及iRMX86段。
- 检查目标、任务、改变、段组合及扩展部。
- 测试和/或交换绝对内存地址的内容。
- 设置、改变、观察并删除断点。
- 观察引起断点的任务表和从表中去掉任务。
- 说明设置断点的任务。
- 测试和/或改变断点任务寄存器的值。
- 设置、改变、观察和删除指定的变量。
- 退出调试程序。

出于以下三方面的原因而采用iRMX86调试程序：

① iRMX86调试程序认识同核和核目标相关的数据结构，而其它的调试程序只能测试内存数据。iRMX86调试程序可以解释与核相关的数据提供出iRMX86目标表以及关于各个目标的详细信息。因此，用户不必通过复杂的数据结构的分类而得到目标信息。

② iRMX86调试程序允许用户在其它任务连续运行时操作或测试一个任务。使用iRMX86调试程序，遇到一个断点时，只停止一个单任务（或者是单作业中的一个任务），其它任务可以继续运行。这样，用户可在多任务环境中进行调试。

③ iRMX86调试程序允许用户监督系统情况而不妨碍其执行。例如，用户可以在邮箱中设置断点，使调试程序显示每次送到邮箱或从邮箱取走的信息。

用户可以从用户iRMX86终端打入CTRL/D而调用调试程序。调试程序用下列信息进行响应：

```
iRMX86 DEBUGGER V3.0
```

“*”是调试程序的提示符。它表示调试程序准备接收终端输入命令。

一个任务可以将调试程序做为故障管理程序。当任务发生故障时，调试程序将显示故障信息。任务可用下列方法把调试程序做为自己的故障管理程序：

- 通过使用SET/EXCEPTION\$ HANDLER系统调用。
- 通过将调试程序做为缺有故障管理程序。当用户连接调试程序做为缺有故障管理程序时, 可使用%SYSTEM宏。
- 当使用CREATE\$ JOB或CREATE\$ TASK建立任务时, 可以说明调试程序做为故障管理程序。

8. 终端管理程序

终端管理程序支持操作员终端和在iRM86核控制下运行的任务之间的实时、异步 I/O。它通常用于不包含iRMX86 I/O系统的应用场合。

终端管理程序有如下特点:

- 行编辑功能。
- 键盘控制, 包括输出挂起和恢复, 以及删除由任务送到终端的数据。
- 响应输入到终端管理程序的行缓冲器中去的字符。

当使用一个在终端管理程序控制下的终端时, 操作员或是由终端显示读一个输出信息, 或是在终端键盘上输入字符。通常, 输入字符是送给任务的输入信息。特殊的输入字符可直接使终端管理程序获得特殊功能。特殊字符有RUBOUT、CR、LF、ESC、CTRL/C、CTRL/O、CTRL/Q、CTRL/R、CTRL/S、CTRL/X、CTRL/Z。终端管理程序的输出没有任何特殊字符。

下面是特殊字符和它的功能说明:

RUBOUT;

内部意义: 将最后打入的但没有删除的字符从当前行中删除。如果当前行是空的, 无内部意义。

外部意义: 如果当前行是空的, BEL (07H) 字符送到终端。否则, 字符按照两种擦模式之一被擦除。在拷贝模式, 字符从当前行删除时也影响显示。如: 在打入“CAT”后再按压RUBOUT键三次, 则显示“CATTAC”。在空格模式, 删除的字符用空格字符显示在GRT屏幕上。如: 在打入“CAT”后再按压RUBOUT键三次, 则从显示中删除所有三个字符。拷贝模式是缺省模式。缺省模式也可在构成应用系统时改变。

CR、LF、ESCape: 都是输入行结束信号。

内部意义: 放ASCII传输结束字符(OAH)或ESCape字符(IBH)在当前行, 这两个信号都表示信息结束。输入信息被送到相应的邮箱。

外部意义: 如果行结束表示符是CR或LF, 则CR和LF都被送到终端。如果表示符是ESCape, 对显示无影响。

CTRL/C;

该字符调用一个用户写的过程。这个过程必须在COMPACT或SMALL控制下编译。它可以形成任何适于应用的功能。通常它使一个应用程序夭折。

CTRL/O;

如果在通常模式下输出, CTRL/O将其变为抑制模式。如果输出在抑制模式, CTRL/O将其变为通用模式。如果输出在编队模式, CTRL/O不起作用。就内部来说, 当输出被抑制时, 任务传送的信息被返回到任务, 就象输出没有被抑制一样。

CTRL/S;

将输出置于编队模式。

CTRL/Q:

与CTRL/S的作用相反。它允许送到邮箱的输出显示，被抑制的输出重新显示。如果要制止其输出显示可重新按压CTRL/S。

CTRL/X:

内部意义：空出当前行。

外部意义：顺序将#、CR、LF送到终端。

CTRL/Z:

内部意义：将当前被处理的输入申请信息的ACTUAL区域置0。然后将信息送到相应的响应邮箱。

无外部意义。

1-2 iRMX86 PC系统的硬件环境

iRMX86操作系统是专门为OEM (Original Equipment Manufactures) 用户设计的操作系统。OEM用户可由此而构成各种适于同要求的应用系统。iRMX86操作系统具有多种多样的功能，功能齐全必然结构庞大，但在具体的应用系统中，可以只具有几种用户要求的功能而把不需要的内容去掉。iRMX86 PC操作系统就是INTEL专门设计用于开发和运行程序的一个通用操作系统。它的运行需要特定的硬件环境。图1-3表示了iRMX86 PC操作系统运行时所需要的硬件环境。

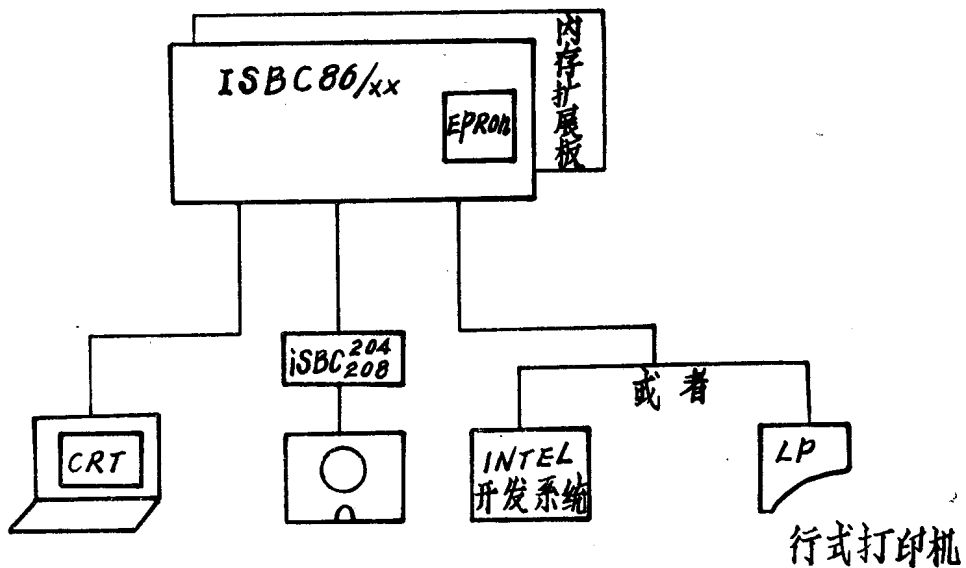


图 1-3

iRMX86、硬件环境

为了应用iRMX86 PC操作系统需要下列硬件部件：

1. INTEL iSBC86/12A或iSBC86/14或iSBC86/30单板计算机。

2. iSBC208软盘控制器及最少两台软盘驱动器。（iSBC208软盘控制器可以联接四台软盘驱动器）。

3. 一台CRT终端。

4. 相应的电源及机箱。

此外，可以联接一台行式打印机或iSBC957B软件包。iSBC957B软件包允许用户把系统直接联到一个INTELLEC微处理机开发系统上去。运行操作系统时既不需要行式打印机，也不需要iSBC957B。

图1—4表示运行iRMX86 PC操作系统时，计算机内存的分布情况。

自由空间区域可运行用户程序和系统应用程序。图中“？”号表示在用户系统中选择多少自由空间。为了运行iRMX86命令大约需要32KB自由内存空间。如运行编译程序则需要更大的内存空间。

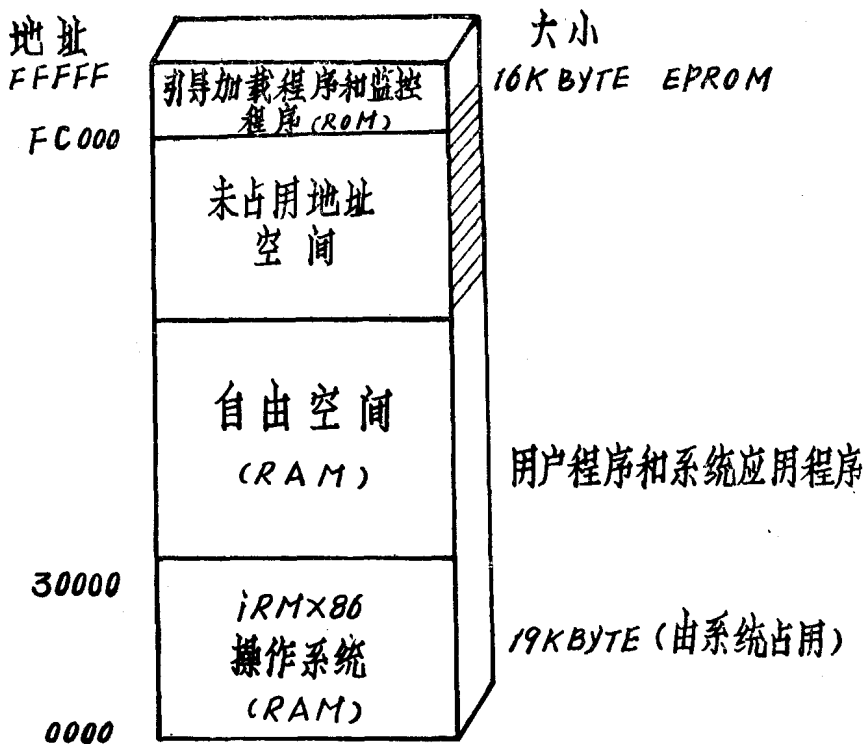


图1—4 iRMX86系统内存分配图

1—3 语言翻译程序和应用程序

为了开发程序，需要语言翻译程序和应用程序。这些语言翻译和应用程序允许编译或汇编程序、连接程序、定位程序地址、建立程序库以及转换绝对目标模块为十六进制格式等等。它们包括：

1. EDIT：标准iRMX86编辑程序。
2. ASM86：8086/8087/8088宏汇编程序。
3. PLM86：PL/M-86编译程序。
4. LINK86：8086连接程序，它把各个编译完的目标模块组合为一个可定位目标模块。
5. LOC86：8086定位程序，它为可定位目标模块按排绝对地址。
6. LIB86：8086库程序，它建立和维修目标模块库。
7. OH86：转换绝对目标模块为十六进制格式的程序。
8. PASCAL-86：PASCAL编译程序。
9. FORTRAN-86：FORTRAN编译程序，它与已存在的FORTRAN-80码兼容，也包含FORTRAN-77语言特点。

利用这些程序可以建立能由终端调用的可执行程序。

关于这些软件产品的详细说明请参阅各自的产品手册。

1—4 通用开发接口 (UDI)

iRMX 86 PC操作系统提供一个通用的开发接口 (UDI)。UDI为用户程序和INTEL软件提供了一个使用操作系统的标准方法。UDI可以被认为是“软件总线”，UDI系统调用的设置相当于硬件系统中总线的约定。利用UDI软件有三个重要的优点：

1. 对操作系统的改变有相对的独立性：UDI系统调用的设置在操作系统中保持稳定不变，这样，系统中的软件，不管是用户开发的软件还是直接安装的软件都可以保持完整无损。

2. 简单：利用UDI“软件总线”，语言翻译程序、应用程序或任何其他仅用UDI系统调用同基础操作系统进行通讯的软件包都能够被设置在任何支持UDI的操作系统上。

3. 独立的卖主软件：UDI系统允许软件卖主单独地提供多种在iRMX86操作系统上运行的程序。

第五章将详细讨论如何在iRMX 86 PC操作系统上应用UDI系统调用。

1—5 iRMX86文件系统

iRMX86操作系统的基本功能是提供一个文件系统。用户程序也和语言翻译程序、应用程序一样需要建立和删除文件、打开和关闭文件、读和写文件，也需要形成另外一些文件操作。这些文件存放在信息存贮设备中（如软盘）。下面讨论iRMX86文件系统的主要特征。

1. 文件的等级命名：用户通过终端使用命令操作文件。程序用系统调用处理文件。iRMX86操作系统允许用户系统编组它的命名文件成树状结构。如图1—5所示。

这个等级结构由文件和目录组成。图中三角形代表文件，矩形代表目录。这种结构方式允许信息逻辑编组并用最上部存取。

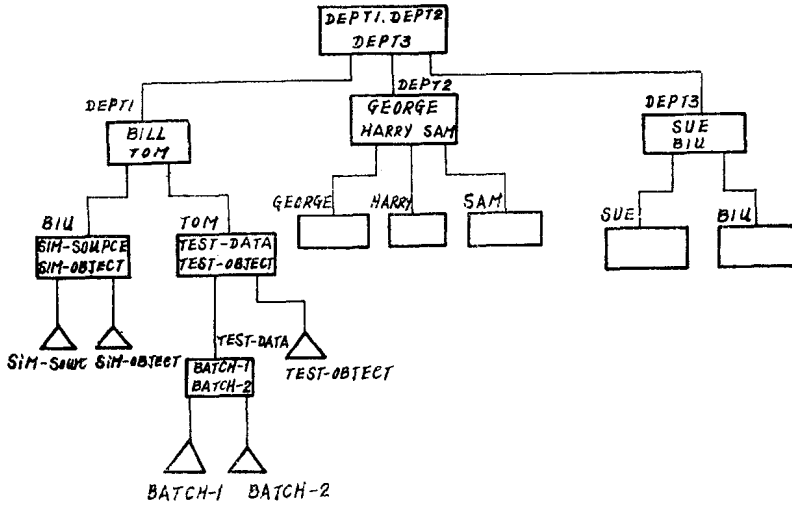


图1-5

等级文件结构

2. iRMX86文件术语

下面是用于描述iRMX86文件系统的一些术语的意义：

- 文件：在一个计算机系统中，一个文件简单地说就是通过一个名字而集合的相关信息。文件驻留在象盘一类的第二存储器中。一个iRMX86文件名是路径名的最后一部分，通过路径名可显式或隐式地表示文件所属的存储媒体和目录。

- 卷：卷是第二存储设备，如软盘、硬盘、磁泡存储器。这些存储设备在接受文件和目录之前必须格式化。格式化使用人机接口中的FORMAT命令。

- 目录：目录用以编辑（即逻辑组合并定位）文件和其他目录。不能象在文件中那样直接在目录中存放信息，而是由操作系统管理目录信息，增加、删除或修改目录条目。象文件一样，目录由路径名表示。文件和目录必须包含在同一个卷的特定目录中。

- 路径名：通常，系统调用和命令都用路径名表示一个文件或目录。路径名可以认为是一个文件或目录的准确的名字。一个路径名描述目录树上的一条通路。它不仅说明文件或目录的名字，而且也表明文件和目录所属的实际目录和设备。

在图1-4中，名字为“BILL”的目录是不准确的，因为存在有两个“BILL”目录名。但路径名“DEPT1/BILL”则是准确的，因为它是唯一的。（“/”字符划分路径名的各元素）路径名“DEPT1/TOM/TEST-OBJECT”是一个文件的有效标识符。

- 逻辑名：逻辑名用以标识设备，同时它由“:”号划定边界。例如，用 :LP: 表示行式打印机。对物理设备赋予逻辑名的命令是“ATTACHDEVICE”。

iRMX86PC操作系统已经赋予逻辑名的设备有：

- CI：控制台输入（终端键盘）。

: CO : 控制台输出 (终端屏幕)。

: LP : 行式打印机的逻辑名。

: BB : 字节槽。这不是一个实际的物理设备。写到 : BB : 的任何东西都不显示, 而是由 : BB : 返回一个 EOF (end-of-file)。

: AFDO : iSBC208控制的软盘驱动器O的逻辑名。

: AFDI : iSBC208控制的软盘驱动器I的逻辑名。

3. 终端文件操作

用户通过终端用程序运行操作iRMX86文件和目录。程序由后跟参数的单字命令来调用。iRMX86操作系统提供了形成操作的程序, 这在开发系统中通常是必须的。

终端文件操作包括:

- COPY 用以拷贝文件。
- DIR 用以显示目录内容。
- RENAME 给文件或目录一个新名字。
- CREATDIR 用以建立一个新的文件目录。
- SUBMIT 自动执行包含在一个文件中的命令。

这些命令在后面的几章中起详细叙述。

4. 程序文件操作

程序必须能操作文件。例如, 一个汇编程序必须打开和阅读源文件, 也必须建立和写目标文件。用户程序将读、写、删除文件。用户程序借助于UDI系统调用完成这些操作。除了文件操作以外, 用UDI系统调用还可得到其他一些操作系统服务。

5. iRMX86PC操作系统提供的文件和目录

iRMX86PC软件包包括两个软盘:

一个系统盘和一个库盘。系统盘上写有iRMX86操作系统, 库盘上是iRMX86接口库文件。同时, 还提供四片EPROM片子, 上面写有监控程序和引导加载程序。四片EPROM将按装在用户iAPX86单板计算机上。

• 系统盘: 图1-6表示iRMX86CP系统盘的文件结构。这个文件结构的各个元素简要说明如下。

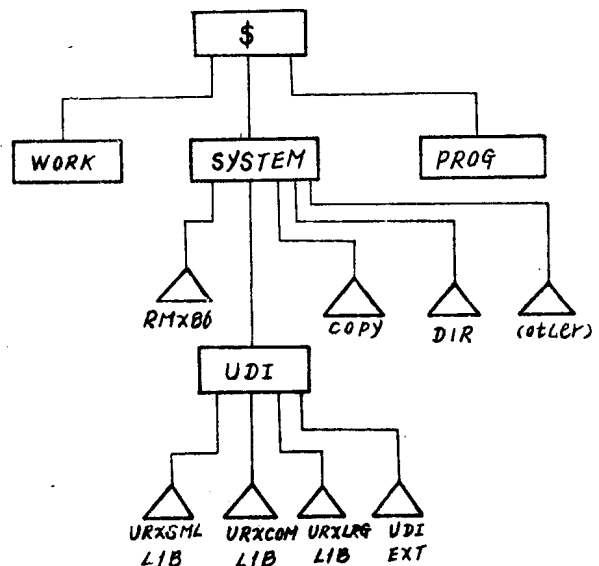


图1-6

iRMX86文件和目录结构