

IBM PC/XT • 286 • 386

微机

汇编语言与 外设编程

王士元 编著

南开大学出版社

IBM PC/XT、286、386 微机

汇编语言与外设编程

王士元 编著

南开大学出版社

[津]新登字 011 号

IBM PC/XT、286、386 微机
汇编语言与外设编程
王士元 编著

南开大学出版社出版

电话:34 9318 邮政编码:300071

(天津八里台南开大学校内)

新华书店天津发行所发行

河北昌黎印刷厂印刷

1993 年 2 月第 1 版

1993 年 2 月第 1 次印刷

开本:787×1092 1/16

印张:24.25 插页:2

字数:677 千字

印数:1—11000

ISBN 7—310—00502—3/TP·11 定价:18.50 元

前 言

汇编语言是和计算机硬件结合最密切的语言,它也是用户能够使用的最快和最能发挥硬件功能的唯一的一种计算机语言。在实际的许多应用程序中,多要和显示器、键盘、打印机、磁盘、异步串行通信接口打交道,并且需要直接控制它们,因而只能使用汇编语言来对这些外围设备进行编程,本书从基本汇编语言基础开始,结合 PC/XT、286、386 硬件,循序渐进,结合实例,使读者可一边看书,一边上机实践,从而比较容易地对 PC 机的外围设备有所了解,并且逐步地掌握 8088(80286、80386)汇编程序设计方法,以及对外围设备的编程技术。

由于微机技术的高速发展,目前高分辨显示器用的愈来愈多,因此对驱动这些显示器的适配器如 EGA(增强的图形适配器)和 VGA(视频图形阵列)的编程技术,是许多人想了解的,可惜目前这方面的书很少,因此本书对 EGA 和 VGA 的结构及编程作了较系统的介绍,并给出了一些编程实例,这些内容可作为对 EGA 和 VGA 编程的基础。

目前 PC 286、PC 386 及 PC 486 微机正在迅速地占领微机市场,虽然它们的功能在不断增强,档次在不断升高,然而 CPU 均属同一系列,8088 称为 16 位的 CPU,80286 为 16 位的,80386 为 32 位的,它们的指令是向上兼容的,或可以说它们的指令集是 8088 的高级集,为此本书对 80286 和 80386 增加的指令和增强了功能的指令均给出了说明,并有相应的应用实例。

若要编程使用这些指令,尤其要能使用 80386 的新指令,就得使用 Microsoft 的 MASM 5.0 版的宏汇编(当然还有其它类似的宏汇编),为此本书对该版本新的特色作了介绍,这些特色使得对汇编编程更方便更灵活,功能更强。

作者力求将最近几年来国内外市场上出现的较新的知识介绍给读者,然而遗憾的是这方面的资料和书籍并不多见,但愿本书对读者能有所帮助。

本书开始的内容适合汇编语言初学者使用,但对有汇编语言编程能力的人也很适合,尤其后面的内容,对已能熟练进行汇编编程的人,是必须要了解的。

本书是硬件与软件结合,但以软件为主,它可作为高校与计算机有关专业的计算机应用与汇编语言程序设计的教科书或参考书,也可供科研和工程技术人员搞科研开发使用或参考。

书中所附的程序,均录入在一张软盘中,对需要的读者可以提供。作者也可提供多种接口卡,如 A/D、D/A,以及许多较新的 I/O 卡,以提供科研和教学用。

作 者

1991 年于南开园

目 录

1 概述

2 基本概念

- 2.1 微型机的基本部分..... (10)
- 2.2 二进制、十六进制和 ASCII 码 (11)
- 2.3 二进制和十六进制数算法..... (15)
- 2.4 在主存储器中存储信息..... (16)
- 2.5 中央处理器的工作..... (16)
- 2.6 为什么需要汇编语言..... (17)

3 8088 的结构

- 3.1 8088 寄存器组 (19)
- 3.2 存储器段地址..... (20)
- 3.3 8088 指令集 (23)
 - 3.3.1 数据寻址方式..... (25)
 - 3.3.2 堆栈操作..... (25)
 - 3.3.3 I/O 和其它数据传送指令 (28)
 - 3.3.4 算术指令和标志寄存器..... (28)
 - 3.3.5 逻辑指令..... (30)
 - 3.3.6 字串处理指令..... (32)
 - 3.3.7 控制转移指令..... (37)
 - 3.3.8 循环控制指令..... (42)
 - 3.3.9 子程序调用和返回指令..... (44)
 - 3.3.10 中断及中断指令 (46)
 - 3.3.11 处理器控制指令 (48)

4 BIOS、DOS 和宏汇编

- 4.1 开机..... (50)
- 4.2 运行用户程序..... (51)
- 4.3 伪指令..... (52)

4.3.1	定义数据的伪指令.....	(53)
4.3.2	定义过程的伪指令和过程的属性.....	(54)
4.3.3	段的定义伪指令和寻址.....	(54)
4.4	DOS 连接用户程序的方法	(56)
4.5	一个程序例子.....	(57)
4.6	建立程序.....	(62)
4.7	BIOS 的子程序	(63)
4.8	汇编操作符.....	(64)
4.9	宏指令.....	(65)
4.9.1	宏定义和宏调用.....	(66)
4.9.2	PURGE 伪操作	(66)
4.9.3	LOCAL 伪操作	(67)

5 PC 系统板

5.1	总线概念.....	(68)
5.2	内存的支持器件.....	(68)
5.3	系统支持器件.....	(71)
5.3.1	8259 中断控制器	(72)
5.3.2	8255 可编程外围接口	(73)
5.3.3	键盘.....	(74)
5.3.4	8253 定时器/计数器	(82)
5.3.5	产生发声效果的程序.....	(87)

6 单色、彩色/图形和打印机适配器

6.1	单色显示器.....	(97)
6.1.1	显示字符和属性的程序.....	(99)
6.1.2	6845 CRT(阴极射线管)显示控制器	(104)
6.1.3	6845 内部寄存器和光标的控制	(106)
6.1.4	方块移动程序	(109)
6.1.5	显示器驱动程序	(112)
6.2	彩色字符/图形显示器适配器.....	(118)
6.2.1	字符方式	(119)
6.2.2	屏幕卷动的程序	(122)
6.2.3	图形方式	(128)
6.2.4	一个画图程序	(131)
6.3	并行打印机适配器	(136)
6.3.1	驱动并行打印机的过程	(137)
6.3.2	打印字符的子程序	(140)
6.3.3	用 DOS 打印功能调用打印字符串的程序.....	(141)

7 EGA 和 VGA 彩色显示适配器

7.1 工作方式	(148)
7.2 EGA/VGA 的结构	(151)
7.3 显示存储器	(152)
7.3.1 文本方式时 VRAM 的使用	(152)
7.3.2 图形方式时 VRAM 的使用	(155)
7.4 图形控制器	(160)
7.4.1 锁存器	(160)
7.4.2 逻辑单元	(160)
7.4.3 图形控制器中的寄存器	(161)
7.5 CRT 控制器	(167)
7.5.1 几个重要的寄存器	(168)
7.6 定序器	(172)
7.7 属性控制器	(174)
7.8 外部寄存器	(177)
7.9 小结	(179)
7.9.1 屏幕上某象素点(x,y)对应于 VRAM 的偏移地址计算方法	(181)
7.9.2 写 VRAM 的方法	(181)
7.9.3 读 VRAM 的方法	(187)
7.9.4 保存屏幕图象的程序	(188)
7.9.5 显示图象文件的程序	(191)
7.9.6 画一段水平线	(194)
7.10 BIOS 显示功能调用	(198)
7.10.1 BIOS 显示基本功能调用	(199)
7.10.2 EGA/VGA 扩充的 BIOS 功能调用	(204)
7.11 BIOS 数据区	(221)

8 异步串行通信

8.1 串行和并行通信	(222)
8.2 异步串行通信的规约	(223)
8.3 通用异步接收发送器(8250 UART)	(224)
8.4 调制解调器(MODEM)	(226)
8.5 RS232C 接口标准	(228)
8.6 BIOS 的异步串行通信功能调用	(229)
8.7 8250 的内部寄存器	(231)
8.8 8250 的初始化	(232)
8.8.1 设置波特率	(232)
8.8.2 设置通信数据格式	(233)
8.8.3 MODEM 控制寄存器的初始化	(233)
8.8.4 设置操作方式	(233)
8.8.5 设置中断允许寄存器	(235)

8.9 用 8250 进行通信	(235)
8.10 8250 中断	(236)
8.11 简单的查询式通信程序	(237)
8.12 中断式通信程序	(241)

9 磁盘的读和写

9.1 DOS 控制下的软盘读写操作	(250)
9.1.1 文件控制块 FCB 和数据传输区 DTA	(252)
9.1.2 DOS 对磁盘文件进行读写时的功能调用	(254)
9.2 顺序存取文件	(256)
9.2.1 写一个顺序文件	(256)
9.2.2 顺序读一个文件	(257)
9.2.3 顺序存取文件例	(258)
9.3 随机存取方式	(262)
9.4 随机块存取方式	(264)
9.4.1 随机块读和写的过程	(265)
9.4.2 随机块读程序例	(267)
9.5 文件代号式存取方式	(270)
9.5.1 文件代号式存取的过程	(270)
9.5.2 对磁盘文件进行文件代号式存取程序例	(274)
9.6 用 BIOS 功能调用对软盘进行读写	(279)
9.6.1 对磁盘进行读写的 BIOS 功能调用	(279)
9.6.2 BIOS 读磁盘根目录程序例	(281)

10 宏汇编 MASM 5.0

10.1 段的简化定义	(286)
10.1.1 简化段定义的伪指令	(287)
10.1.2 定义段的类型	(287)
10.1.3 等价名的使用	(289)
10.1.4 段组定义伪指令	(289)
10.1.5 定义内存模式伪指令	(290)
10.1.6 段名的缺省名	(291)
10.2 设置 MASM 功能的伪指令	(292)
10.2.1 定义段次序的伪指令	(292)
10.2.2 使汇编产生特定处理器指令的伪指令	(293)
10.2.3 使用 IEEE 格式对实数进行编码的伪指令	(293)
10.2.4 使信息输出到标准输出设备上伪指令 %OUT	(295)
10.3 用简化段定义建立一个程序框架	(295)
10.3.1 标准程序框架	(295)
10.3.2 一个 .COM 格式的程序	(297)
10.3.3 串等值的使用	(298)

10.4	关于过程的定义.....	(298)
10.4.1	标准的过程定义.....	(298)
10.4.2	简化的过程定义.....	(299)
10.4.3	在过程中利用堆栈传递参数.....	(301)
10.5	模块化程序设计.....	(301)
10.5.1	符号说明伪指令 PUBLIC 和 EXTRN	(303)
10.5.2	一个简单的模块结构例.....	(305)
10.5.3	关于 INCLUDE 文件.....	(306)
10.5.4	COMM 伪指令	(307)
10.5.5	指定一个库文件伪指令 INCLUDELIB	(308)
10.6	条件汇编.....	(308)
10.6.1	条件汇编伪指令.....	(309)
10.6.2	条件错伪指令.....	(310)
10.7	MASM 的选择项	(312)
10.7.1	MASM 的选择项	(312)
10.7.2	汇编一个文件.....	(315)
10.7.3	读汇编列表文件.....	(315)
10.7.4	交插参考列表文件.....	(316)
10.7.5	用 Code Veiw debugger 调试程序	(319)
10.8	MASM 5.0 版新的特色总结	(319)

11 80286 和 80386 CPU 与增加的新指令

11.1	80286 和 80386 的操作方式	(322)
11.1.1	实地址方式.....	(322)
11.1.2	保护虚地址方式.....	(322)
11.2	80286 的基本结构	(325)
11.2.1	寄存器.....	(326)
11.2.2	中断系统.....	(327)
11.3	80286 增强与增加的指令	(328)
11.3.1	使用堆栈的指令.....	(328)
11.3.2	带符号整数乘法指令.....	(328)
11.3.3	移位指令.....	(328)
11.3.4	内存范围检查的指令.....	(329)
11.3.5	记录栈空间的指令.....	(329)
11.3.6	撤消栈空间的指令.....	(329)
11.3.7	字符串输入指令.....	(330)
11.3.8	字符串输出指令.....	(331)
11.3.9	控制保护态指令.....	(331)
11.4	80386 的结构特点	(332)
11.5	80386 新增加的指令	(332)

11.5.1	测试与置位指令.....	(332)
11.5.2	位扫描指令.....	(334)
11.5.3	数的传送与扩展指令.....	(335)
11.5.4	多位移动指令.....	(335)
11.5.5	条件设置指令.....	(336)
11.6	80386 增加了功能的指令	(336)
11.6.1	整数乘指令.....	(336)
11.6.2	转换指令.....	(336)
11.6.3	字符串操作指令.....	(337)
11.6.4	32 位栈操作指令	(337)
11.6.5	中断返回指令.....	(337)
11.7	在 DOS 下如何使用 80386 新指令	(338)
11.8	使用 80386 微处理器指令编程例.....	(338)
附录 1 8088、80286、80386 指令集.....		(341)
附录 2 汇编错误代码注释		(371)
1.	错误代码.....	(371)
2.	未编号的错误信息.....	(381)

目 录

1 概述

2 基本概念

- 2.1 微型机的基本部分..... (10)
- 2.2 二进制、十六进制和 ASCII 码 (11)
- 2.3 二进制和十六进制数算法..... (15)
- 2.4 在主存存储器中存储信息..... (16)
- 2.5 中央处理器的工作..... (16)
- 2.6 为什么需要汇编语言..... (17)

3 8088 的结构

- 3.1 8088 寄存器组 (19)
- 3.2 存储器段地址..... (20)
- 3.3 8088 指令集 (23)
 - 3.3.1 数据寻址方式..... (25)
 - 3.3.2 堆栈操作..... (25)
 - 3.3.3 I/O 和其它数据传送指令 (28)
 - 3.3.4 算术指令和标志寄存器..... (28)
 - 3.3.5 逻辑指令..... (30)
 - 3.3.6 字串处理指令..... (32)
 - 3.3.7 控制转移指令..... (37)
 - 3.3.8 循环控制指令..... (42)
 - 3.3.9 子程序调用和返回指令..... (44)
 - 3.3.10 中断及中断指令 (46)
 - 3.3.11 处理器控制指令 (48)

4 BIOS、DOS 和宏汇编

- 4.1 开机..... (50)
- 4.2 运行用户程序..... (51)
- 4.3 伪指令..... (52)

4.3.1	定义数据的伪指令.....	(53)
4.3.2	定义过程的伪指令和过程的属性.....	(54)
4.3.3	段的定义伪指令和寻址.....	(54)
4.4	DOS 连接用户程序的方法	(56)
4.5	一个程序例子.....	(57)
4.6	建立程序.....	(62)
4.7	BIOS 的子程序	(63)
4.8	汇编操作符.....	(64)
4.9	宏指令.....	(65)
4.9.1	宏定义和宏调用.....	(66)
4.9.2	PURGE 伪操作	(66)
4.9.3	LOCAL 伪操作	(67)

5 PC 系统板

5.1	总线概念.....	(68)
5.2	内存的支持器件.....	(68)
5.3	系统支持器件.....	(71)
5.3.1	8259 中断控制器	(72)
5.3.2	8255 可编程外围接口	(73)
5.3.3	键盘.....	(74)
5.3.4	8253 定时器/计数器	(82)
5.3.5	产生发声效果的程序.....	(87)

6 单色、彩色/图形和打印机适配器

6.1	单色显示器.....	(97)
6.1.1	显示字符和属性的程序.....	(99)
6.1.2	6845 CRT(阴极射线管)显示控制器	(104)
6.1.3	6845 内部寄存器和光标的控制	(106)
6.1.4	方块移动程序	(109)
6.1.5	显示器驱动程序	(112)
6.2	彩色字符/图形显示器适配器.....	(118)
6.2.1	字符方式	(119)
6.2.2	屏幕卷动的程序	(122)
6.2.3	图形方式	(128)
6.2.4	一个画图程序	(131)
6.3	并行打印机适配器	(136)
6.3.1	驱动并行打印机的过程	(137)
6.3.2	打印字符的子程序	(140)
6.3.3	用 DOS 打印功能调用打印字符串的程序.....	(141)

7 EGA 和 VGA 彩色显示适配器

7.1	工作方式	(148)
7.2	EGA/VGA 的结构	(151)
7.3	显示存储器	(152)
7.3.1	文本方式时 VRAM 的使用	(152)
7.3.2	图形方式时 VRAM 的使用	(155)
7.4	图形控制器	(160)
7.4.1	锁存器	(160)
7.4.2	逻辑单元	(160)
7.4.3	图形控制器中的寄存器	(161)
7.5	CRT 控制器	(167)
7.5.1	几个重要的寄存器	(168)
7.6	定序器	(172)
7.7	属性控制器	(174)
7.8	外部寄存器	(177)
7.9	小结	(179)
7.9.1	屏幕上某象素点(x,y)对应于 VRAM 的偏移地址计算方法	(181)
7.9.2	写 VRAM 的方法	(181)
7.9.3	读 VRAM 的方法	(187)
7.9.4	保存屏幕图象的程序	(188)
7.9.5	显示图象文件的程序	(191)
7.9.6	画一段水平线	(194)
7.10	BIOS 显示功能调用	(198)
7.10.1	BIOS 显示基本功能调用	(199)
7.10.2	EGA/VGA 扩充的 BIOS 功能调用	(204)
7.11	BIOS 数据区	(221)

8 异步串行通信

8.1	串行和并行通信	(222)
8.2	异步串行通信的规约	(223)
8.3	通用异步接收发送器(8250 UART)	(224)
8.4	调制解调器(MODEM)	(226)
8.5	RS232C 接口标准	(228)
8.6	BIOS 的异步串行通信功能调用	(229)
8.7	8250 的内部寄存器	(231)
8.8	8250 的初始化	(232)
8.8.1	设置波特率	(232)
8.8.2	设置通信数据格式	(233)
8.8.3	MODEM 控制寄存器的初始化	(233)
8.8.4	设置操作方式	(233)
8.8.5	设置中断允许寄存器	(235)

8.9 用 8250 进行通信	(235)
8.10 8250 中断	(236)
8.11 简单的查询式通信程序	(237)
8.12 中断式通信程序	(241)

9 磁盘的读和写

9.1 DOS 控制下的软盘读写操作	(250)
9.1.1 文件控制块 FCB 和数据传输区 DTA	(252)
9.1.2 DOS 对磁盘文件进行读写时的功能调用	(254)
9.2 顺序存取文件	(256)
9.2.1 写一个顺序文件	(256)
9.2.2 顺序读一个文件	(257)
9.2.3 顺序存取文件例	(258)
9.3 随机存取方式	(262)
9.4 随机块存取方式	(264)
9.4.1 随机块读和写的过程	(265)
9.4.2 随机块读程序例	(267)
9.5 文件代号式存取方式	(270)
9.5.1 文件代号式存取的过程	(270)
9.5.2 对磁盘文件进行文件代号式存取程序例	(274)
9.6 用 BIOS 功能调用对软盘进行读写	(279)
9.6.1 对磁盘进行读写的 BIOS 功能调用	(279)
9.6.2 BIOS 读磁盘根目录程序例	(281)

10 宏汇编 MASM 5.0

10.1 段的简化定义	(286)
10.1.1 简化段定义的伪指令	(287)
10.1.2 定义段的类型	(287)
10.1.3 等价名的使用	(289)
10.1.4 段组定义伪指令	(289)
10.1.5 定义内存模式伪指令	(290)
10.1.6 段名的缺省名	(291)
10.2 设置 MASM 功能的伪指令	(292)
10.2.1 定义段次序的伪指令	(292)
10.2.2 使汇编产生特定处理器指令的伪指令	(293)
10.2.3 使用 IEEE 格式对实数进行编码的伪指令	(293)
10.2.4 使信息输出到标准输出设备上伪指令 %OUT	(295)
10.3 用简化段定义建立一个程序框架	(295)
10.3.1 标准程序框架	(295)
10.3.2 一个 .COM 格式的程序	(297)
10.3.3 串等值的使用	(298)

10.4 关于过程的定义.....	(298)
10.4.1 标准的过程定义.....	(298)
10.4.2 简化的过程定义.....	(299)
10.4.3 在过程中利用堆栈传递参数.....	(301)
10.5 模块化程序设计.....	(301)
10.5.1 符号说明伪指令 PUBLIC 和 EXTRN	(303)
10.5.2 一个简单的模块结构例.....	(305)
10.5.3 关于 INCLUDE 文件.....	(306)
10.5.4 COMM 伪指令	(307)
10.5.5 指定一个库文件伪指令 INCLUDELIB	(308)
10.6 条件汇编.....	(308)
10.6.1 条件汇编伪指令.....	(309)
10.6.2 条件错伪指令.....	(310)
10.7 MASM 的选择项	(312)
10.7.1 MASM 的选择项	(312)
10.7.2 汇编一个文件.....	(315)
10.7.3 读汇编列表文件.....	(315)
10.7.4 交插参考列表文件.....	(316)
10.7.5 用 Code View debugger 调试程序	(319)
10.8 MASM 5.0 版新的特色总结	(319)

11 80286 和 80386 CPU 与增加的新指令

11.1 80286 和 80386 的操作方式	(322)
11.1.1 实地址方式.....	(322)
11.1.2 保护虚地址方式.....	(322)
11.2 80286 的基本结构	(325)
11.2.1 寄存器.....	(326)
11.2.2 中断系统.....	(327)
11.3 80286 增强与增加的指令	(328)
11.3.1 使用堆栈的指令.....	(328)
11.3.2 带符号整数乘法指令.....	(328)
11.3.3 移位指令.....	(328)
11.3.4 内存范围检查的指令.....	(329)
11.3.5 记录栈空间的指令.....	(329)
11.3.6 撤消栈空间的指令.....	(329)
11.3.7 字符串输入指令.....	(330)
11.3.8 字符串输出指令.....	(331)
11.3.9 控制保护态指令.....	(331)
11.4 80386 的结构特点	(332)
11.5 80386 新增加的指令	(332)

11.5.1	测试与置位指令.....	(332)
11.5.2	位扫描指令.....	(334)
11.5.3	数的传送与扩展指令.....	(335)
11.5.4	多位移动指令.....	(335)
11.5.5	条件设置指令.....	(336)
11.6	80386 增加了功能的指令	(336)
11.6.1	整数乘指令.....	(336)
11.6.2	转换指令.....	(336)
11.6.3	字符串操作指令.....	(337)
11.6.4	32 位栈操作指令	(337)
11.6.5	中断返回指令.....	(337)
11.7	在 DOS 下如何使用 80386 新指令	(338)
11.8	使用 80386 微处理器指令编程例.....	(338)
附录 1	8088、80286、80386 指令集	(341)
附录 2	汇编错误代码注释	(371)
1.	错误代码.....	(371)
2.	未编号的错误信息.....	(381)

概 述

1981年8月美国IBM公司推出了称为**IBM PC**的个人计算机,这种微型计算机与过去的机型相比有许多重要的改进,最重要的是使用了Intel公司生产的8088微处理器作为PC机的中央处理器。

虽然8088是一个8位的芯片,但其结构和16位芯片8086一样,因而PC机就能够使用大容量的内存储器;大部分微型机仅能用64K内存,而PC机却能高达1024K(或称1M的内存)。利用这个特点,IBM公司将一个大而有用的BASIC语言解释程序固化在只读存储器ROM中,该ROM是1M内存的一小部分,因此许多人在使用PC机时,用BASIC语言来编写程序。当然更主要的原因是,BASIC语言易学易懂,编程者无需对硬件环境有所了解。现在更好的一些BASIC语言已经在流行,如BASICA、True BASIC、GWBASIC、quick BASIC等,但使用BASIC语言也有不足之处,下面来讨论这个问题。

BASIC语言易学而且好用,但它也有一些缺点。BASIC编程者把PC看作是能够执行BASIC语句和函数的一种计算机,但是实际上,PC只能够执行由中央处理器8088提供的功能,这些功能由8088机器指令来指示完成。机器指令比BASIC语句更多更基本,例如BASIC中的PRINT语句,实际上由上百条8088机器指令来完成,因此PC机执行PRINT语句,实际上是执行了上百条的8088机器指令。当一个BASIC程序被执行时,每一个语句必须要翻译成许多机器指令,然后执行这些机器指令,即一个语句的执行,实际上是执行了代替它的一系列的机器指令,这个过程如图1-1所示,称为边解释边执行,当然执行速度就相当慢了。

如果我们编写一个仅由机器指令组成的程序,那么它就可以直接由8088中央处理器来执行,这种程序的执行时间就要比BASIC程序快许多倍,这是因为它省掉了每个程序语句进行翻译的过程,而且用BASIC许多语句完成的功能。仅需几条机器指令就可实现,如图1-2所示。由机器指令组成的程序是用汇编语言写的。用汇编语言写8088机器指令程序的语言,称为8088汇编语言。

如前所述,BASIC编程者把计算机看作是一个BASIC机器,在这个机器上只能使用BASIC语言所提供的一些功能。但是使用汇编语言编程,不仅程序执行速度快,而且能够发挥出每个硬件的功能。在PC机上运行用汇编语言编的程序时,实际上是控制了使机器工作的每个电路。仿佛机器对用户是透明的,可观察到电路的动作。