

大 型 计 算 机 丛 书

IBM S/390 汇编语言

王 勇 周 曦 马光志 编著

浙江大学出版社

大型计算机丛书

IBM S/390 汇编语言

王 勇 周 曦 马光志 编著

浙江大学出版社

图书在版编目 (CIP) 数据

IBM S/390 汇编语言 / 王勇, 周曦, 马光志编著.
杭州: 浙江大学出版社, 2000. 8
(大型计算机丛书)
ISBN 7-308-02407-5

I. I... II. ①王...②周...③马... III. 汇编语言-程序设计 IV. TP313

中国版本图书馆 CIP 数据核字 (2000) 第 34008 号

责任编辑: 邹小宁

装帧设计: 俞亚彤

出版发行: 浙江大学出版社

(杭州浙大路 38 号 邮政编码 310027)

(网址: <http://www.zjupress.com>)

(E-mail: zupress@mail.hz.zj.cn)

排 版: 浙江大学出版社电脑排版中心

印 刷: 余杭市人民印刷有限公司

经 销: 浙江省新华书店

开 本: 787mm×1092mm 1/16

印 张: 16.25

字 数: 416 千字

版、印次: 2000 年 8 月第 1 版 2000 年 8 月第 1 次印刷

印 数: 0001—2700

书 号: ISBN 7-308-02407-5/TP·197

定 价: 40.00 元

内 容 简 介

本书以 IBM S/390 大型计算机为背景,介绍了汇编语言程序设计的基本概念、基本原理和一般方法。其主要内容有:S/390 的地位和作用、S/390 的体系结构和硬件特征、程序设计的一般概念、汇编语言与汇编程序、顺序与分支程序设计、循环程序设计、子程序、十进制运算、宏汇编等等。另外,本书还介绍了汇编语言程序设计的一些基本方法和技巧。本书可作为高等学校计算机系应用数据处理专业(大型机专业)的教材和教学参考书,也可供 IBM S/390 机的维护人员和有关技术人员阅读参考。

前 言

1997年夏,教育部(原国家教委)与美国国际商用机器公司(IBM)开始合作了一项投资超过3000万美元的教育项目,IBM向中国四所重点大学(北京大学、复旦大学、华中理工大学、华南理工大学)捐赠四台IBM最先进的大型服务器系统S/390,教育部在这四所高校的计算机系开设一个新的专业(方向)——应用数据处理(大型机)。本书即是根据此专业(方向)的教学计划与教学大纲编写而成的。

本书以IBM S/390大型计算机为背景,以S/390汇编语言为工具介绍了汇编语言程序设计的基本概念、基本原理和一般方法。其内容包括:S/390概述、S/390汇编语言、程序设计的一般概念、顺序与分支程序、循环程序、子程序、十进制运算、宏汇编等等。为了使读者更快地掌握程序设计的方法和更方便地进行程序调试,本书中采用了ASSIST汇编程序提供的一些宏指令。

全书共分为十章。第1章为绪论,介绍了S/390计算机系统的发展、地位和作用以及汇编语言的特点。第2章介绍了学习汇编语言需要了解的一些基本知识,如S/390的体系结构、寻址方式、指令类型与格式、汇编语言的数据和基本成分等。第3章介绍了汇编程序工作的概念以及汇编控制指令。第4章介绍了顺序与分支程序设计的概念和方法。第5章介绍了循环程序和内部子程序设计。第6章为综合程序设计小结。第7章介绍了外部子程序设计。第8章介绍了十进制运算。第9章介绍了一组有用的高级指令。第10章简单地探讨了宏和条件汇编的概念和方法。在各章之后,均有一定数量的习题,可供读者练习以便掌握本书的内容。附录为IBM S/390系统和汇编语言的有关资料,可供读者查阅。

本书在编写上是以程序设计为导向的,在介绍完S/390的基本概念和硬件特征以及汇编语言的基本概念之后,没有专门的章节完整地介绍指令系统,而是直接进入程序设计阶段,根据程序的需要介绍有关的指令,这样做的目的是为了读者能更快更好地掌握程序设计方法。

本书可作为高等学校计算机系应用数据处理专业(大型机专业)的教材和教学参考书,也可供IBM S/390机的维护人员和有关技术人员阅读参考。

本书吸取了我们多年来在汇编语言程序设计方面的工作/教学实践的一些体

NJSS/07 - 1 -

会和成果,并曾以讲义的形式在华南理工大学计算机工程与科学系九五、九六、九七级学生中讲授过。在编写过程中参考了 IBM 的技术资讯、国外有关教材和国内兄弟院校的有关资料,并得到了华中理工大学、复旦大学计算机系以及 IBM 中国公司大学合作部的全力支持和帮助,在此表示衷心的感谢。

由于我们的水平有限和时间比较仓促,书中的错误和不妥之处在所难免,敬请读者批评指正,不胜感激。

作 者

2000 年 2 月于华南理工大学

目 录

第1章 绪论

1.1 IBM 大型机的历史及发展	1
1.2 S/390 的地位和作用	2
1.2.1 计算机系统的种类及特点	2
1.2.2 S/390 的地位和作用	2
1.3 汇编语言的特点	3
1.3.1 汇编语言的特点	3
1.3.2 掌握机器特性的必要性	4
1.3.3 汇编语言进行程序设计的一般过程	4
习题一	5

第2章 准备知识

2.1 S/390 体系结构及硬件特征	6
2.1.1 中央处理器	7
2.1.2 主存储器	8
2.1.3 扩充存储器	9
2.1.4 通道子系统	9
2.1.5 辅助存储器	10
2.1.6 系统构造	10
2.2 S/390 的寻址方式	11
2.2.1 寄存器寻址方式	11
2.2.2 立即数寻址方式	11
2.2.3 存储器寻址方式	11
2.3 S/390 指令的格式与类型	13
2.3.1 S/390 指令格式	13
2.3.2 S/390 指令类型	14
2.4 S/390 汇编语言的数据表示	15
2.4.1 字符数据	15
2.4.2 二进制数	15
2.4.3 逻辑数	16
2.4.4 十进制数	16

2.5 S/390 汇编语言的基本成分	17
2.5.1 基本字符集	17
2.5.2 符号名	17
2.5.3 项及表达式	18
2.5.4 汇编语句的分类和一般格式	19
2.5.5 机器指令的符号形式	20
2.5.6 汇编语言的书写格式	21
习题二	22

第3章 汇编程序和汇编控制语句

3.1 汇编程序 Assembler	23
3.2 汇编控制语句	25
3.2.1 数据与数据区域的定义	25
3.2.2 程序的构成控制	31
3.2.3 地址的指定	35
3.2.4 程序间的结合	37
3.2.5 符号等价语句 EQU	38
3.2.6 产生符号表的 LTORG 语句	38
3.2.7 拷贝语句 COPY	39
3.2.8 打印控制 PRINT	40
习题三	41

第4章 顺序与分支程序设计

4.1 程序的功能	42
4.2 几条简单指令介绍	43
4.2.1 RR(寄存器—寄存器)型指令	43
4.2.2 RX(寄存器—变址存储器)型指令	44
4.3 完整的程序示例	45
4.4 程序的改进	46
4.4.1 内存转储指令 XDUMP	47
4.4.2 隐式地址的使用	47
4.5 条件码的设置与分支程序设计	47
4.6 数据的输入和转换	50
4.6.1 数据的输入 XREAD 指令	50
4.6.2 输入数据的转换 XDECI 指令	52
4.7 数据的输出和转换	53

4.7.1 字符数据的输出 XPRNT 指令	54
4.7.2 输出数据的转换 XDECO 指令	55
4.8 更多的指令和程序示例	56
4.8.1 加法指令 A	56
4.8.2 减法指令 S	56
4.8.3 寄存器比较指令 CR	57
4.8.4 比较指令 C	57
4.8.5 装载补码指令 LCR	60
4.8.6 装载负数指令 LNR	61
4.8.7 装载正数指令 LPR	61
4.8.8 装载测试指令 LTR	61
4.8.9 存储字符指令 STC	64
4.8.10 装载字符指令 IC	64
4.9 乘法和除法指令	64
4.9.1 寄存器乘法指令 MR	65
4.9.2 乘法指令 M	65
4.9.3 寄存器除法指令 DR	66
4.9.4 除法指令 D	66
4.10 寄存器等价和扩展助记符	69
4.10.1 寄存器等价的使用	69
4.10.2 扩展助记符的使用	70
4.11 符号常数的使用	70
4.12 取地址指令的使用	73
习题四	76

第5章 循环程序及内部子程序设计

5.1 循环程序设计	83
5.1.1 计数循环指令 BCT	85
5.1.2 计数循环寄存器指令 BCTR	85
5.1.3 小于等于循环指令 BXLE	90
5.1.4 大于循环指令 BXH	92
5.2 内部子程序设计	93
5.2.1 分支连接指令 BAL	93
5.2.2 分支连接寄存器指令 BALR	102
5.2.3 多寄存器保存指令 STM	102
5.2.4 多寄存器恢复指令 LM	103

5.2.5 A 型地址常量	103
习题五	104

第 6 章 综合程序设计

6.1 程序文档的标准	110
6.2 存储器—立即数类(SI)指令示例	111
6.2.1 移动立即数指令 MVI	112
6.2.2 立即数逻辑比较指令 CLI	112
6.3 存储器—存储器类(SS)指令示例	113
6.3.1 移动字符串指令 MVC	113
6.3.2 逻辑比较指令 CLC	115
6.4 地址计数器的访问	115
6.4.1 引用地址计数器的值	116
6.4.2 控制地址计数器的值	117
6.5 逻辑运算和比较指令	126
习题六	128

第 7 章 外部子程序设计

7.1 外部子程序介绍	135
7.2 外部子程序设计	136
7.2.1 外部子程序的结构和调用方式	136
7.2.2 外部子程序的编写细节	138
7.2.3 小结	147
7.3 与高级语言的连接	148
7.3.1 从高级语言调用汇编子程序	148
7.3.2 从汇编语言调用高级语言子程序	149
7.4 虚拟段的使用	149
习题七	154

第 8 章 十进制运算

8.1 压缩和解压指令	157
8.1.1 压缩指令 PACK	158
8.1.2 解压指令 UNPK	159
8.2 十进制加减运算	160
8.2.1 十进制加法指令 AP	161
8.2.2 十进制减法指令 SP	161

8.2.3 清零及加十进制指令 ZAP	161
8.2.4 比较十进制指令 CP	162
8.3 十进制乘除运算	163
8.3.1 十进制乘法指令 MP	163
8.3.2 十进制除法指令 DP	163
8.4 输出编辑	164
8.5 十进制移位操作	181
8.6 十进制与二进制之间的转换	182
8.6.1 转换二进制指令 CVB	182
8.6.2 转换十进制指令 CVD	182
8.6.3 转换的标准技术	183
习题八	183

第9章 高级指令介绍

9.1 逻辑运算指令	189
9.1.1 逻辑与指令 N	190
9.1.2 寄存器逻辑与指令 NR	190
9.1.3 逻辑或指令 O	191
9.1.4 寄存器逻辑或指令 OR	191
9.1.5 逻辑异或指令 X	192
9.1.6 寄存器逻辑异或指令 XR	192
9.1.7 立即数逻辑与指令 NI	193
9.1.8 立即数逻辑或指令 OI	193
9.1.9 立即数逻辑异或指令 XI	193
9.1.10 内存逻辑异或指令 XC	195
9.1.11 内存逻辑或指令 OC	195
9.1.12 内存逻辑与指令 NC	196
9.2 屏蔽测试运算指令	196
9.3 移位操作	197
9.3.1 逻辑左移单寄存器指令 SLL	198
9.3.2 逻辑右移单寄存器指令 SRL	198
9.3.3 逻辑左移双寄存器指令 SLDL	199
9.3.4 逻辑右移双寄存器指令 SRDL	199
9.3.5 算术左移单寄存器指令 SLA	200
9.3.6 算术右移单寄存器指令 SRA	201
9.3.7 算术左移双寄存器指令 SLDA	201

9.3.8 算术右移双寄存器指令 SRDA	201
9.4 半字指令	205
9.4.1 装入半字指令 LH	206
9.4.2 存储半字指令 STH	206
9.4.3 比较半字指令 CH	206
9.4.4 加半字指令 AH	207
9.4.5 减半字指令 SH	207
9.4.6 乘半字指令 MH	207
9.5 MVCL,CLCL 和 EX 指令	211
9.5.1 长移动指令 MVCL	211
9.5.2 长逻辑比较指令 CLCL	212
9.5.3 执行指令 EX	213
9.6 TR 指令	213
9.7 TRT 指令	216
9.8 CLM,ICM 和 STCM 指令	222
9.8.1 字符屏蔽逻辑比较指令 CLM	222
9.8.2 屏蔽插入字符指令 ICM	223
9.8.3 屏蔽存储字符指令 STCM	223
9.9 SPM 指令	224
习题九	224

第 10 章 宏及条件汇编

10.1 符号参数和宏	230
10.1.1 宏的定义及其与子程序的比较	230
10.1.2 符号参数	231
10.1.3 宏定义及宏调用	231
10.2 条件汇编	236
10.2.1 AIF 指令	236
10.2.2 AGO 指令	237
10.2.3 ANOP 指令	237
习题十	239

附录 A 扩展助记符表	242
附录 B IBM EBCDIC 编码表	243
附录 C 常用指令表	245
参考文献	248

第 1 章 绪 论

1.1 IBM 大型机的历史及发展

1966 年 4 月,IBM 公司宣布了 System/360(以下简称 S/360)系列计算机,这在计算机发展史上具有划时代的意义。在此之前,所有的计算机都是专门设计和制造的,仅用于军事、科研等部门,相互之间是不通用的;而 S/360 是世界上第一台批量生产的通用型计算机,价格便宜(相对于专用机型而言)而且可以广泛地应用于科学和工程计算、商业及事务处理等多种领域,所以很快得到了市场的认同,逐渐占据了世界上大、中型通用计算机市场的统治地位,得到了业界广泛的应用。这之后,IBM 继续不断地改进和发展改型计算机,70 年代推出了 System/370,80 年代推出了 System/370-XA(Extended Architecture)和 ES9000(Enterprise System),90 年代推出了 System/390。与此同时,在 S 系列大型机上的操作系统也从最初的简单单道作业系统发展到了今天世界上规模最大、功能最强、最复杂的大型操作系统 MVS/OS390。

图 1.1 是该系列计算机的发展过程示意图。

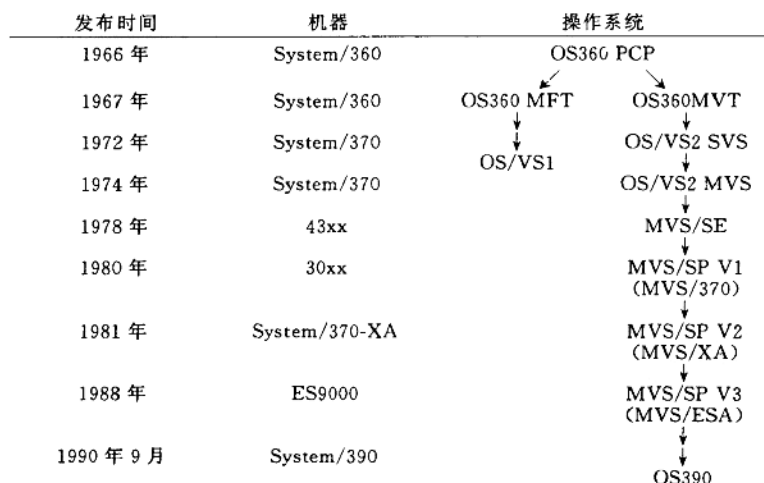


图 1.1 IBM S 系列大型通用计算机的发展过程示意图

图中:MFT——Multiprogramming with a Fixed number of Tasks;
MVT——Multiprogramming with a Variable number of Tasks;
SVS——Single Virtual Storage;
MVS——Multiple Virtual Storage;
ESA——Enterprise System Architecture。

1.2 S/390 的地位和作用

1.2.1 计算机系统的种类及特点

在计算机产业界,一味地肯定或否定大型计算机的作用是不明智的。正如每种程序设计语言都有自己的特点和应用领域一样,不同的计算机也有自己的特点和应用领域,没有任何一种单一的计算机系统类型可以适合于所有的应用领域。在今天的计算机产业界,主要存在着以下几种类型的计算机系统:

(1)以 S/390 为代表的大中型计算机系统

这种类型的计算机系统主要技术特点:CPU 利用率极高,多任务处理水平高,具有密集的 I/O 处理,能很好地处理不同类型的工作负载,并且提供非常高的可靠性和可提供性。由于这些技术特点,传统上这种类型的计算机系统主要用于处理大中型企业的商业计算任务。

(2)以 Unix 机器为代表的中小型计算机系统

这种类型的计算机系统主要技术特点:面向计算,多任务处理水平高,具有中等程度的 I/O 处理,在相同类型的工作负载下能很好地处理,可靠性和可提供性一般。由于这些技术特点,传统上这种类型的计算机系统主要用于工程计算和中小企业的商业计算。另外,这种类型的计算机系统经过改良之后的高端类型可用于大规模的科学计算。

(3)以 PC 为代表的微型计算机系统

这种类型的计算机系统主要技术特点:面向单个的用户,友好的图形界面,多任务处理水平低,I/O 处理不密集,可靠性和可提供性较差。传统上这种类型的计算机系统主要用于与计算机系统的最终用户交互,在这种平台上提供有相当丰富的应用程序,例如文字处理、电子表格、游戏等。

1.2.2 S/390 的地位和作用

传统上,对于企业一级的应用而言,拥有强大处理能力的大型计算机系统的优势是不言而喻的,但是,该系统也有缺点,主要是体积庞大、价格昂贵、维护费用高昂、软件不够丰富等等。70 年代末诞生的微型计算机以及更早一些诞生的中小型计算机系统随着其性能在 80 年代的大大提高和价格的大大降低,开始挑战大型计算机系统在企业一级应用的统治地位。分布式处理、通用平台(Unix)、小型微型化开始在企业界流行和普及,在 80 年代末大型计算机系统的地位已经开始岌岌可危,甚至有人提出了“恐龙论”,认为大型计算机系统是行将灭绝的恐龙,很快将会完全被中小型

和微型计算机系统所取代。

真的是这样吗? 90 年代计算机工业的发展否定了这一预测,因为在八九十年代这二十年中,前十年的发展以计算机系统小型微型化为主要特征,而后十年的发展则以计算机系统网络化为主要特征,在这一趋势的推动下,有三个主要的动因挽救了大型计算机系统的生命并使之茁壮地成长:

①计算机用户方面。随着业务的成长,采用小型微型计算机系统的企业需要不断地增加新的系统才能适应业务的需要,他们逐渐发现,购置一个庞大的分布式系统所花的投资也许比同等规模的大型计算机系统要便宜,然而要维护一个庞大的分布式系统所花费的费用却要远远高于同等规模的大型计算机系统,并且前者的可靠性和可提供性要远低于后者。

②IBM 方面。在新的公司总裁的领导下,开始了振兴大型计算机系统的努力。首先从硬件方面着手,采用了小型微型计算机系统常用的 CMOS 技术来制造 CPU 芯片,机器的发热量大大减少,从而可以采用普通的风冷系统代替价格昂贵的水冷系统,计算机系统的价格和运行费用因此大大降低;软件方面,在增强传统应用的同时,大型计算机系统开始支持流行的应用,例如 Unix 和 NT 应用,并开始提供新的功能,例如,支持 Java 开发工具,群件,Web 服务,等等。

③整个产业界方面。以 Internet, Intranet, Extranet 为代表的计算机网络技术的流行和普及,推动了对网络服务器的需求。在迅猛增长的网络客户需求的要求下,以小型和微型计算机系统充当的网络服务器在响应性能、安全性和可靠性方面都已经不能满足需求了。怎样解决这一问题? 一是采用多台小型微型机器组成群集系统,但是群集系统不便于管理,并且也不大安全可靠;二是采用大型计算机系统来充当超级服务器,从而很好地解决了管理和安全性、可靠性的问题。

由于以上三个方面的原因,从 90 年代中期开始,计算机界开始了所谓“再集中”的发展趋势,即企业应用重新集中到以 S/390 为代表的大型服务器当中去。据第三方信息评估机构的估计,今天全世界 80% 以上的信息存储在 IBM 的大型服务器当中。

1.3 汇编语言的特点

1.3.1 汇编语言的特点

汇编语言是与机器密切相关的一种程序设计语言,属于第二代程序设计语言。在汇编语言出现之前,人们使用机器语言(亦即第一代程序设计语言)进行编程。

机器语言使用二进制来代表指令和数据,是计算机裸机能够直接理解的惟一语言。用机器语言编写程序可以充分利用机器指令的灵活性,达到精确的控制和较高的效率。但是这种程序使用的二进制编码形式比较难读难写,不便于维护和修改,掌握和使用这种语言的难度相当大。

为了解决这个问题,人们开始使用汇编语言进行程序设计。汇编语言的主要思想使用简单易记的符号(例如英文单词)来表示机器指令,即使用“助记符”代替机器指令的操作码,使用“标识符”代替地址。与机器语言相比,在程序效率牺牲不大的情况下,程序的可读性和可维护性大大增强。因此,在底层程序设计领域,汇编语言已经成为最主流的语言。

在高级语言广泛使用的今天,汇编语言的使用仍然是不可缺少的。因为使用汇编语言编程可以充分利用机器的硬件特性,编写出来的程序代码占用空间小,效率高,所以在编写一些系统程序时

(例如操作系统内核、硬件驱动程序),汇编语言是第一选择。另外,在要求实时响应和精确控制的工业现场,汇编语言也是得力的工具语言。

1.3.2 掌握机器特性的必要性

由于汇编语言是与机器密切相关的一种程序设计语言,因此在进行汇编语言程序设计时,必须了解机器的有关特性,这样才能编出正确的汇编语言程序。概括起来,我们所要了解的机器特性有:

①存储器的基本单元、容量。机器存储器的基本单元(即编址的基本单位)是什么?例如字节。以相邻字节为单位进行操作的单元有哪些种类?例如字、半字、双字等。其字长是多少?存储容量多大(即基本单元数量是多少)?地址表示的方法和地址变换机构(用于支持虚拟存储器)又是怎样的?

②控制器的功能及相互关系。必须了解的有:通用寄存器(包括累加器)的数量、功能、程序状态字(包括状态寄存器和指令地址计数器)与程序设计有关的性能等问题。

③数据类型和格式。机器能处理哪些类型的数据?例如定点数、浮点数、十进制数、字符数据等。每种数据在机器内部的表示方法(即机内格式)是怎样的?

④指令格式、长度、功能及寻址方式。对机器指令系统的深入了解是汇编语言程序设计的重要任务之一,它包括:指令格式是怎样的?指令的长度为何?指令中操作数的地址是怎样形成的(即寻址方式如何)?指令的种类有哪些?每一条指令的功能如何?程序设计应注意些什么?等等。

⑤控制器的其他特性。汇编语言程序设计要了解算术逻辑部件和控制器的状态(这些信息集中反映在所谓程序状态字 PSW 中)、中断、时钟等。有些机器的控制器允许有多对互斥状态。例如,目态与管态,等待与运行,停机与工作等。所谓目态就是运行用户程序(或问题程序)的状态,在此状态下禁止使用特权(或管态)指令。而管态是管理程序运行的状态,在这种情况下,可以运行包括特权指令在内的所有指令。中断也是十分重要的特性,简单说来,中断就是当发生某种紧急或意外事件时,需要算术逻辑部件和控制器停止执行当前的程序而去处理该事件,处理完后,再去执行被暂时停止执行的程序或转去执行其他程序。因此,需要了解哪些中断与程序设计有关?中断时的现场保护在什么地方?如何屏蔽与开放中断?机器的时钟如何利用?等等。

⑥外部设备的特性。外部设备是计算机与外部世界交换信息的手段。进行汇编语言程序设计时,必须了解有哪些外部设备可供使用?每种外部设备的具体使用方法如何(包括信息交换的机构和交换方式)?如何使外部设备与算术逻辑部件和控制器协调工作(包括速度匹配)?有无通道设施?如何组织通道程序?等等。

1.3.3 汇编语言进行程序设计的一般过程

程序设计包括了多方面的工作,特别是解决复杂问题时情况更是如此。因此,程序设计一般要经历以下几个阶段:

(1)定义问题

这个阶段也称为要求定义分析,或称需求分析,即对要解决的问题的意义和要求,了解明白准确,这包括制订一系列清晰而无二义性的规格说明。例如问题要求什么样的精度,提供的输入是什

么,以及期望的结果又是什么,等等。这个阶段是十分重要的,对复杂问题尤其如此。因为它产生的规格说明书是以后各阶段的依据。如果不把问题的含义弄明白、弄准确,那将会导致整个设计的失败或返工。

(2)构造解法概要

这个阶段也称为功能设计。主要是制定整个解法的概要,即总体设计。这包括将整个问题分解成若干任务或子任务(这主要是按功能划分)以及它们之间相互关系的描述。目前采用的有功能模块分割法、逐步求精法等。

(3)确定算法

这个阶段也称为结构设计。主要是选择最优的算法和数据结构以实现上述的每一个任务(或子任务),即根据功能模块而选择最适当的方法和数据结构以实现之。这是程序设计的核心步骤,也是比较困难的工作。

(4)编码

这个阶段也称为制作程序或编程序。即把算法用此汇编语言书写出来。

(5)调试和排除出错

这个阶段亦称测试。主要是通过测试来保证程序正确。一般要作静态检查和动态检查。静态检查包括人工检查和上机进行语法检查。动态检查就是根据程序在工作中所有可能的情况(例如对各种不同初始数据的输入)进行测试,检查执行是否正确。这个阶段的工作量一般是比较大的。

(6)整理文档

程序设计的结果包括两大部分,一是程序,二是文档。文档包括各个设计阶段的规格说明书以及用户使用手册等,这是用户使用、维护程序的依据。

(7)维护

这主要是保持程序能正常运行(在用户的整个使用期间)。这包括程序运行中发生错误后对程序的修改,以及当条件、环境发生变化和增加某些功能时,需要修改规格说明和相应的程序。这就是说,它包括程序正确性、完善性、适应性的维护。

习 题 一

1. 计算机系统有哪些种类?各有什么特点?
2. S/390 为什么在计算机产业界有着极其重要的地位和作用?
3. 为什么要学习汇编语言?
4. 学习汇编语言要掌握哪些机器特性?
5. 用汇编语言开发应用的过程和步骤是什么?