

D.S. 董 芳

COBOL 语言基础教程

结构化的程序设计方法

科学出版社

1981年10月第1版

COBOL 语言基础教程

——结构化的程序设计方法

科学出版社

TP312

D61

237520

COBOL 语言基础教程

——结构化的程序设计方法

D. S. 董 著

胡承慈 孙嘉鸿 居融融 译

肖业伦 校



科学出版社

1986

55180/10/

内 容 简 介

本书叙述在带有磁盘操作系统的IBM370计算机上实现的ANS COBOL, 采用了结构化程序设计的概念。内容较系统, 详细, 由浅入深。书中例题丰富多样, 并已实际上机运行过。本书可供高等院校有关专业师生及企业、事业单位管理人员学习、参考。

Douglas S. Tung

ELEMENTARY COBOL: A STRUCTURED PROGRAMMING APPROACH

The Chinese University Press (HONG KONG), 1981

COBOL 语言基础教程

——结构化的程序设计方法

D. S. 董 著

胡承慈 孙嘉鸿 屠融融 译

肖业伦 校

责任编辑 杨家福 刘晓融

科学出版社出版

北京朝阳门内大街137号

中国科学院开封印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

1986年1月第一版 开本: 787×1092 1/16

1986年1月第一次印刷 印张: 21 1/2

印数: 0001—6,600 字数: 492,000

统一书号: 15031·689

本社书号: 4429·15—8

定价: 5.05 元

译 者 的 话

COBOL 是一种高级程序设计语言,是为处理大量数据而设计的。它除了可以进行数字处理,还可以进行文字资料处理。COBOL 应用范围很广,不仅能用于商业,也能用于工业、农业、交通运输业、军事部门和政府部门等;不仅能用于经济管理工作,也能用于人事管理和图书管理工作等。凡日常所遇到的各种库存管理、工资管理、劳动定额管理、银行业务、各种资料和报表的汇总与统计加工,以及情报检索工作等等,无不属于它的应用范围。

COBOL 语言比其它高级程序设计语言更接近于英语口语,并且符合企业管理人员的使用习惯,因而用它编写出来的程序人们比较容易了解,易读性好。同时,由于它具有处理大量数据的能力和它的程序通用性好,易于从一种计算机更换到另一种计算机上运行,因而使它成为目前世界上使用最广泛的语言之一,占各种语言上机总量的 50—70%。

当然,这种语言也存在一定的缺点,诸如:语言文本的内容比较庞大,不便于记忆,不容易学习;用它编制出来的程序较长,不精炼。正因为这样,我们就更加需要一本较好的教材和一套题材广泛的源程序样本供我们学习、参考、模仿。本书的优点是内容较系统、详细,由浅入深,例题丰富多样,并已上机运行过。本书在香港中文大学作为电子计算机系二年级学生 COBOL 语言课程的主要教科书已有几年,使用效果较好。

目前,随着我国计算机事业的发展,COBOL 语言正在逐步得到运用。我们相信,作为科学管理工作中的一种极有效的工具,COBOL 的使用必将越来越广泛。因此,我们翻译了本书。

在翻译和校对过程中,我们改正了一些认为不够清楚的地方,并对此作了注释和补充,个别的地方则予以删去。

由于译者的水平有限,译文难免有翻译不妥或错误之处,敬请读者批评指正。

序 言

本书是为便于学习商业程序设计语言 COBOL (Common Business Oriented Language) 而编写的。COBOL 已作为计算机在工业、商业和政府机构管理应用上的标准程序设计语言。这种语言在数据系统语言会议(CODASYL)的倡议下不断地得到精炼和修改。该会议机构是一个由很多计算机制造厂商和用户组成的工作委员会。目前,ANS COBOL 被认为是工业界的标准,而目前流行的版本则是 1974 标准 COBOL*。

本书叙述在带有磁盘操作系统的 IBM 370 计算机上实现的 ANS COBOL。它依据的是 1968 COBOL 标准,有少数地方参考了 1974 COBOL 标准。

COBOL 的设计意图是为便于学习和使用,但对于面向科学的学生来说,它有一些在任何程序设计语言中都能找到的最复杂的语句。为使 COBOL 语句的应用简单方便,用短小的例题加以说明。每一章都有一个或几个实例程序用于说明本章包含的内容。这些实例取自商业数据处理的多种领域:简单的工资报表,库存情况报表,库存周转率计算,销售登记,输入数据的有效性验证,顺序文件、直接组织文件和索引顺序文件的更新。此外,有些程序与科学上的应用有关。所有程序都已经上机运行过,并列出其源(程序)清单。

为了适应程序设计技巧的最新发展,本书采用了结构化程序设计的概念。结构化程序是这样一种程序,它使用有限的一组逻辑结构,遵循模块式设计的某些原则,从而简化程序设计工作并改善程序的清晰性。结构化程序设计并不是什么高深的课题,它只不过是设计和发展可靠程序的一种正确途径。从一开始就把这种技巧灌输给学生,使他们养成良好的程序设计习惯,这是极有好处的。

本书假定读者已经通过阅读一本书或参加计算机的短训班而对计算机有所了解。在香港中文大学,要求学生把《计算机原理和程序设计基础》作为第一门课程,在该课程中举例说明了 FORTRAN 的应用。本书可以作为学习 COBOL 的主要教材,或者作为数据处理课程的语言教材,以补充其它介绍计算机科学或信息系统的一般性教材。

本书共有十四章。第一章介绍基本概念。第二章至第五章给出语言的子集,以便学生能够编写完整的程序。可以认为,学生愈早地学会编写和解程序,他就能愈快地懂得这种语言。第六章介绍结构化程序设计的基本概念。其余各章(第七章至第十四章)提供这种语言的综合性内容,包括以下的一些题目,例如内部排序方法,表处理,文件的组织和处理,以及附加的课题如 SORT 动词和子程序等。

作者对在本书准备过程中给予帮助的人们表示诚挚的谢意。特别是对香港中文大学计算机科学系 S. C. Loh 教授, H. S. Hung 博士, S. Hu 博士和 L. Kong 先生给予的支持和鼓励表示感谢;向香港中文大学计算机科学系 C. P. Cheng 先生和英语系 Marilou Covey 小姐以及香港工业学院计算机科学系 L. K. H. Chim 夫人表示感谢,感谢他们对本书手稿提出了有益的批评和建议。我也感谢下列人士在本书编写之前和编写过程中所作出的直接或间接的贡献。他们是 C. S. Hui 先生, Joseph Tu 先生和 Regina

* 指美国标准 X 3.23—1974, 即国际标准 ISO 1989—1978。——译者注

Tsui 夫人。最后还要感谢 T. K. Chuok 先生在程序设计和编辑上给予的帮助，感谢 Teresa Chung 小姐和 Cynthia Kwok 夫人打印了原稿。

我必须着重指出，本人对书中可能仍然存在的任何错误和遗漏负全部责任。对读者给本书提出的任何校正、批评和建议，我都表示感谢。

最后我要感谢香港中文大学和工业学院计算机中心有限公司允许我转载它的技术资料“使用 WATBOL-11S 的一般资料”作为本书的附录 C。

D. S. 董

1978 年 7 月

目 录

序言	vii
第一章 基本概念	1
1.1 什么是计算机?	1
1.2 计算机体系结构	1
1.3 解题过程	2
1.4 适宜用计算机解决的问题	3
1.5 COBOL 语言的历史和发展	3
1.6 COBOL 的编译	5
1.7 COBOL 作为一种程序设计语言	6
1.8 FORTRAN 与 COBOL 的比较	6
第二章 COBOL 程序设计引论	9
2.1 数据的层次结构	9
2.2 COBOL 基础知识	10
2.3 程序的模块式设计	12
2.4 一个抽样程序(程序 1)	14
2.5 COBOL 程序的运行	21
2.6 抽样程序的修改(程序 2)	22
第三章 COBOL 的基本成分	25
3.1 可变数据与常数	25
3.2 COBOL 的标点规则	27
3.3 标志法	27
3.4 基本项与组合项	28
3.5 层号	29
3.6 报表标题的打印	30
3.7 “读一个文件,写一个记录”	34
3.8 VALUE 子句	34
3.9 读入未规定数目的输入记录	34
3.10 程序 5	38
第四章 数据部分的基本内容,算术运算和调试	44
4.1 输入数据的描述	44
4.2 运算字符	44
4.3 PICTURE 字符串的允许长度	45
4.4 层号 77	46
4.5 数据卡片的穿孔	46
4.6 数字编辑	47
4.7 算术动词	48
4.8 COMPUTE 语句	54

4.9	数据的传送	55
4.10	一个简单工资报表问题(程序 6)	57
4.11	调试程序	59
第五章	分支语句	69
5.1	顺序控制动词	69
5.2	GO TO 语句	69
5.3	STOP 语句	70
5.4	ALTER 语句	70
5.5	PERFORM 语句	71
5.6	PERFORM 语句的规则	71
5.7	IF 语句	79
5.8	复合条件	89
5.9	嵌套 IF 语句	92
第六章	结构化程序设计与程序设计标准	101
6.1	结构化程序设计	101
6.2	逻辑控制结构的 COBOL 编码	103
6.3	附加的逻辑结构	105
6.4	结构化程序的编写	107
6.5	程序 10	114
6.6	程序设计和编码标准	119
第七章	标识部分, 设备部分和数据部分	123
7.1	引言	123
7.2	标识部分	123
7.3	设备部分	124
7.4	数据部分	128
7.5	编辑字符	134
7.6	例子	139
第八章	数据表示法及数据部分的其他子句	145
8.1	数据表示法	145
8.2	数值数据的五种类型	146
8.3	SYNCHRONIZED 子句	148
8.4	JUSTIFIED 子句	149
8.5	BLANK WHEN ZERO 子句	150
8.6	REDEFINES 子句	150
8.7	OCCURS 子句和下标法	153
8.8	RENAMES 子句	159
8.9	程序 11	160
第九章	过程部分的常用语句	167
9.1	概述	167
9.2	语句类型	167
9.3	输入-输出动词	168
9.4	带有 VARYING 任选项的 PERFORM 语句	173

9.5	CORRESPONDING 的使用	178
9.6	EXAMINE 语句	181
9.7	程序 14	183
第十章	常用的数据处理逻辑	191
10.1	引言	191
10.2	控制改变	191
10.3	多级控制改变	198
10.4	输入记录的有效化	203
10.5	排序方法	210
10.6	例子	212
第十一章	表处理	219
11.1	一维表	219
11.2	一维表的输出	220
11.3	二维表	221
11.4	三维表	224
11.5	例子	226
11.6	查找表格的方法	231
11.7	SET 语句	235
11.8	使用足标查找表格	236
11.9	表格处理中的效率问题	240
第十二章	文件处理(1)——卡片和磁带文件	243
12.1	文件的本质	243
12.2	文件的存贮媒体	243
12.3	实际的与逻辑的文件和记录	243
12.4	穿孔卡片文件	244
12.5	磁带文件	244
12.6	文件的类型和后备技术	248
12.7	简单顺序文件的更新逻辑	249
12.8	关于文件更新的进一步考虑	262
12.9	可变长度的记录处理	262
12.10	程序 28	263
第十三章	文件处理(2)——磁盘文件	270
13.1	引言	270
13.2	文件组织方法	271
13.3	磁盘的优缺点	274
13.4	使用顺序组织文件时的程序设计	274
13.5	使用直接组织文件时的程序设计	278
13.6	使用索引顺序组织文件时的程序设计	285
13.7	索引顺序文件的处理	287
第十四章	COBOL 语言的附加课题	300
14.1	引言	300
14.2	外部排序的概念	300

14.3 COPY 语句	307
14.4 注解和注释	308
14.5 子程序	309
14.6 报表打印系统	312
14.7 源程序中的调试专用语句	316
附录 A 所使用的计算机设备	319
附录 B ANS COBOL 保留字	321
附录 C 使用 WATBOL-11 的一般资料	323
参考文献	326
索引	327

第一章 基本概念

1.1 什么是计算机?

计算机是一种能够解算问题的电子设备,它在接收数据后,对数据施行预定的运算,并提供这些运算的结果。由于计算机的运算速度快,准确性高,而且功能多种多样,因而,目前已被广泛地应用。与“模拟”一词不同,“数字”这个术语用于描述应用在商业数据处理这一类型的电子计算机。数字计算机对离散的数字位进行处理,并根据其相对位置将数据进行记录;为了存贮的目的,每一个数字和每一个字符都有它们自己的位置。数字计算机可达到很高的精确度,这对于商业数据处理是很重要的。模拟计算机则对连续变量形式的数据,例如以电压或温度代表的数据进行运算,其方式是对数据进行物理处理。它们在航空、航天工业和空间探索方面有广泛的应用。

1.2 计算机体系结构

计算机解决问题和产生信息的方式与一个职员完成其工作的方式是很相似的。在人工数据处理¹⁾系统中,职员完成五种基本功能,如图 1.1 所示。

在一计算机系统中,这些功能由设备完成。无论在一种系统中,基本的功能是

- (1) 输入数据至该系统。
- (2) 处理数据:重新整理输入数据并且处理文件。
- (3) 拟定一些过程,它们指明需要什么数据,何时、何地得到这些数据,如何使用这些数据,并给出指令子程序供中央处理机使用。
- (4) 输出:报告处理运算的结果。

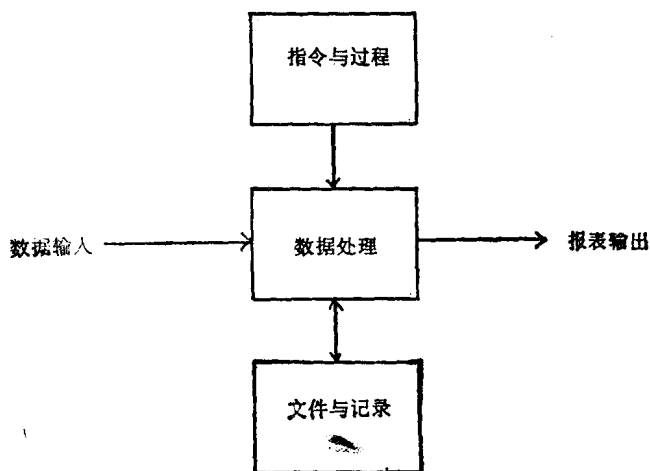


图 1.1 一信息系统的基本组成部分

在计算机系统中,为了把实在的物理设备与发挥该设备用途的程序相区别,称前者为硬件,并称后者为软件。计算机系统的各个部件示于图 1.2 中,并说明如下:

- (1) 把数据输入到计算机处理机的功能由输入设备完成。输入的形式必须是计算机能够接受的。输入到计算机的工具可以是穿孔卡片、穿孔纸带、磁带、电传打字机、键盘、图象显示和电话。

1) 数据处理是商业上使用的术语,意为:在该体系结构内搜集、存贮、处理与传输信息。

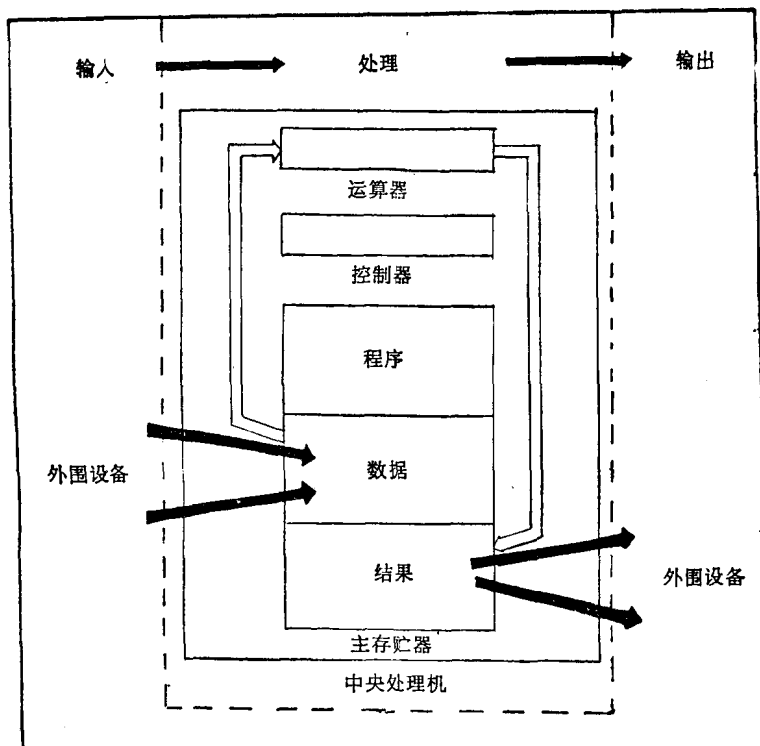


图 1.2 计算机的工作情况

(2) 中央处理机是最重要的部件。它包括:

- (a) 主存储器, 用于存储指令(程序)和正在被运算的数据。
- (b) 算术和逻辑运算器, 用于进行各种计算和比较。
- (c) 控制器, 指导输入设备、输出设备、存储器、算术和逻辑运算器的工作, 使各种处理指令的动作协调一致。

(3) 存储器。

所有数据在被处理之前必须由计算机放在存储器内。存储器包括内存存储器(或称主存储器), 以及作为处理部件部分的外存储器(或称辅助存储器。)存储在外存储器(辅助存储器)中的信息可以传入到主存储器, 或从主存储器传出。

(4) 输出设备生成处理的最后结果。它们把从计算机中获得的信息记录在各种各样的媒体上, 例如穿孔卡片、穿孔纸带、磁带、打印机、阴极射线管、打字机、图象显示终端、绘图仪以及声音转录设备等。

术语“外围”设备表示计算机系统的输入、输出设备与辅助存储器。

1.3 解题过程

在一台计算机能够用来处理数据之前, 必须给它一系列详细的指令, 这一系列指令称为程序。程序设计包括通过下列步骤求解一个给定的问题:

(1) 问题分析——包括对整个问题的理解, 与输入和输出有关的特殊细节的搜集, 要

求进行的计算以及外加的约束条件。

(2) 绘制流程图——用图象表示要求进行的算术和逻辑运算的本质和顺序。它是编写程序文件以备将来参考的一个重要方法。

(3) 编码——将程序的流程图翻译成源程序的算法。

(4) 试算——用试验数据检验程序是否满足所有可能有的条件和情况。而后用实在数据核实该程序。

(5) 调试——根据计算机所提供的对错误的诊断，消除语法上的和编码上的错误。

(6) 文件编制——准备一份程序手册，其中包含对该问题的叙述性的描述、输入数据和输出数据、逻辑以及问题的解答。

程序设计需要技巧和经验。除了不断的实践与改错外没有其它的办法可以替代。

1.4 适宜用计算机解决的问题

计算机绝不是能解决一切问题的万能工具，对一个问题来说用计算机求解也并不总是最有效和最经济的方法。最适宜用计算机求解的问题有下列特征：

(1) 可确定的——该问题可以用有限系列的逻辑和算术步骤清楚地陈述，最好定量处理。

(2) 循环的——运算的性质是重复的，而不是一次性的。

(3) 存贮或计算的容量——容量之大，以致只有计算机化的过程才能承担得起这样的负担。

(4) 成本的评判——由计算机系统生成的信息所获得的利益证实，为了求得该解答而去准备、编写和执行一个程序是值得的。如果一方面处理周期的缩减并不显著，而另一方面在应用更加成熟的方法时，要处理的数据量并不大，则为了加速数据处理而采用费用昂贵的方法是毫无意义的。

1.5 COBOL 语言的历史和发展

商用计算机语言是出于如下愿望而发展起来的，即：希望有一种“接近英语的”、适于商业问题的高级程序设计系统，并作为一种源程序语言，它对任何计算机都是相容的。面向科学的语言，例如 FORTRAN 语言有时也用于数据处理的目的。然而它的结果却是不令人满意的，因为 FORTRAN 并不适于描述和处理大量的数据文件，并且，如果用户不是一个熟练的程序员，它的结构成分和语法不容易被用户理解。

1959年5月28日，由计算机用户、制造厂商和大学代表所组成的一个小组在美国首都华盛顿五角大楼集会，讨论发展一种通用商业语言的问题。这个委员会最早称为 CODASYL (Conference on Data Systems Languages)。这个小组一致通过此项内容，并且制订了发展计划。第一个 COBOL 文本在 1960 年出现。为了赶上计算技术的发展步伐，成立了一个 CODASYL 维护委员会以审查为保持语言的现代化而提出的修改建议。此语言的后续文本出现在 1961, 1964 和 1965 年。1968 年一个标准 COBOL 文本 ANSI

COBOL 为美国国家标准局所制定。ANS COBOL 目前被广泛地应用，本书叙述的就是这个文本。此种标准语言有三个级别的规格，即低级、中级和高级，因此标准 COBOL 可以在磁心尺寸不同的计算机系统上实现。低级 COBOL 可以在存贮器容量为 16 K 的计算机上实现，而高级 COBOL 则可以在 64 K 的计算机系统上实现。低级 COBOL 可以在中级和高级编译程序上进行编译，但反过来则不行。

COBOL 的最新文本是在 1974 年批准的。因为在现有 COBOL 编译程序准备支持这个最新文本的使用之前还需要有一段时间，故在本书中仅提及 1974 ANS COBOL 的某些特有的性质。

如图 1.3 所示，1974 标准 COBOL 分为 12 个功能处理模块。这些模块又可分为两个逻辑组(除报表打印模块外)，即第一级和第二级。第一级是第二级的子集。对每一个功能模块来说，一个 COBOL 执行设备可以执行第一级子集或者第二级的全部性能。一个编译程序如果只执行三个模块的第一级，即核心模块、表处理模块和顺序输入输出模块的第一级，则称为是最低标准的。一个全功能的 COBOL 编译程序执行所有 12 个模块的第二级。很多计算站拥有的编译程序，其执行功能介于最低标准与最高标准之间。

模 块	第 一 级	第 二 级
核 心	最 低 标 准 COBOL	
表 处 理		
顺序输入与输出		
相对输入与输出		
索引输入与输出		
排 序 与 合 并		
报 表 打 印		与第一级相同
分 段		
程 序 库		
调 试		
程 序 间 通 信		
通 信		

图 1.3 1974 标准 COBOL* 的功能模块

1974 ANS COBOL 的模块是

- (1) 核心——进行内部处理的基本语言能力。
- (2) 表处理——定义连续数据项的表，并根据这些数据项在表中的相对位置存取它们的能力。
- (3) 顺序输入与输出——以规定的顺序存取一个文件的记录的能力。
- (4) 相对输入与输出——以随机的形式，或者以顺序的形式存取一个大容量存贮文件的记录的能力。

* 1974 标准 COBOL 即 1974 ANS COBOL。——译者注

(5) 索引输入与输出——以随机的形式, 或者以顺序的形式利用索引存取一个大容量存贮文件的记录的能力。

(6) 排序与合并——根据一组包含在每个记录中由用户规定的键码, 对一个或多个记录文件进行排序, 或对两个或更多个已排序的记录文件进行合并的能力。

(7) 报表打印——由规定报表的物理外观而产生报表的设备。这一设备不规定为产生报表所需的详细过程。

(8) 分段——规定目标程序覆盖要求的能力。

(9) 程序库——指出从程序库中复制正文的能力。

(10) 调试——描述在程序执行期间监控的调试算法的能力。

(11) 程序间通信——一个程序可与一个或多个程序联络的设备。

(12) 通信——存取、处理和产生信息或其中一部分信息的能力。

1.6 COBOL 的编译

计算机懂得用二进制形式(机器语言)表示的指令和数据, 二进制形式(机器语言)是很难学习和跟踪的, 而且编码也很麻烦。用 COBOL 语言编写的程序称为源程序。为了让计算机懂得它, 必须将它翻译成机器语言(目标程序)。翻译工作由一个称为编译程序的程序来完成。不同的计算机有不同的编译程序, 因为它们的机器语言指令是不相同的。编译程序还检查程序在语法上是否正确。在编译过程中发现错误称为诊断。输出装置通常是一台打印机, 将错误返回告诉程序员。程序员必须排除错误以使用机器语言生成没有错误的语句。这一编译过程在图 1.4 上标注为阶段 I。

下一步是把目标程序装入存贮器中, 以便中央处理机执行, 中央处理机读入并且处理数据, 以产生答案回答那个已被编成程序的应用题目。程序的执行在图中标注为阶段 II。程序被执行之后要对结果进行检查, 看其是否正确。如果结果不正确, 则应对源程序或数据加以修正, 重复以上的编译和执行过程, 以便给出正确答案。程序必须考虑到每一个意外事件和一切可能的情况。

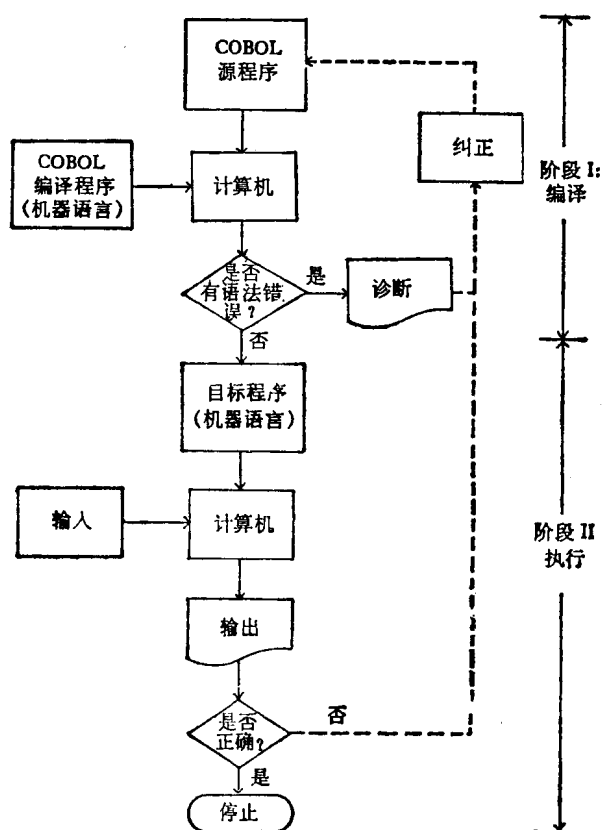


图 1.4 COBOL 编译过程

在上述讨论中, COBOL 编译程序产生目标程序. 对于面向生产的程序所使用的编译程序来说, 这是典型的情况. 然而还有一种在磁心存贮器内的 COBOL 编译程序, 它是专为教学目的而设计的, 例如 WATBOL (Waterloo COBOL), 它的处理工作只需要较短的执行时间. 这些在磁心存贮器内的编译程序生成它们的目标程序, 使其副本直接进入存贮器的另一个区域, 因而可以避免以上所描述的许多操作过程. 此外, 它们还产生更有价值的、执行过程中的错误报文.

1.7 COBOL 作为一种程序设计语言

COBOL 语言得到普遍应用的原因是

(1) COBOL 是唯一的由美国联邦政府支持的语言翻译程序, 美国政府对于拥有一种独立的、高级的、面向商业的计算机语言很感兴趣.

(2) COBOL 具有一种自身即是文件的特征, 即用英语编写的语句易于被非程序编制人员理解.

(3) 通过多次的 CODASYL 会议, 不断地对 COBOL 语言进行了标准化工作, 以增强它对变化所需的和吸收新技术的反应能力.

本语言的一些优点和特征是

(1) COBOL 程序是用精确的、易懂的英语单词和短语编写的. 常用的商业术语被吸收进来以利于与非程序编制人员进行交流. 文件、记录和数据单位的逻辑结构接近于商业实际.

(2) 数据部分与过程部分是互相分开的. 如果输入数据格式必须重写的话, 只要改变数据部分就行了. 同样, 如果处理要求改变的话, 只需在过程部分加以修正即可.

(3) 在 COBOL 编译程序中包含了预先检验过的输入与输出模块, 因而可以免去程序员书写输入与输出格式这种麻烦的事务工作.

COBOL 的缺点是

(1) 为了完成比较简单的任务也需要做较大的程序编制工作. 从工作效率观点来说, FORTRAN, APL 或面向机器的语言是较为优异的.

(2) 在能够编写出一个有意义的程序之前, 学习 COBOL 语言需要熟悉许多细则.

1.8 FORTRAN 与 COBOL 的比较

为了确定这两种语言的优缺点, 用 FORTRAN 和 COBOL 语言编写了四个比较简单

程序类型	源程序中的卡片数目		编译时间(分)		COBOL 比 FORTRAN 编译时间增加的百分比
	FORTRAN	COBOL	FORTRAN	COBOL	
包建工程价格计算	18	72	0.5007	0.7989	59.6
排序子程序	25	56	0.4890	0.8108	65.8
每小时工资表计算	35	115	0.5099	0.8390	64.5
订单处理循环	22	288	0.5040	1.1111	120.46

图 1.5 FORTRAN 与 COBOL 程序的时间比较

的程序并作了试验。这些程序是在由 IBSYS 系统操作的 IBM 7090 计算机上编译的。该系统提供了可比较的编译。图 1.5 列出了试验结果。

由于 COBOL 更接近于用户,很自然地它离机器语言有更远的距离,因而需要更多的编译时间。图 1.6 是 FORTRAN 与 COBOL 的比较表。在初次阅读时并不希望读者能够理解所有的细则,但在完全了解该语言之后,劝告读者再返回来阅读它。

	FORTRAN	COBOL
1. 元素		
空格	空格用于改善程序的可读性,在编译时将被忽略。	空格用作定界符,它对语句的形成是必需的。
句号	句号在浮点常数中使用,它也是关系运算符和逻辑运算符的一部分。	句号是 COBOL 字符集中的一个,用作定界符以及数值常数中的小数点。
保留字	有几个(数学库函数名)。	约 250 个。
数据(变量)名	其类型由第一个字符决定,或由 TYPE 语句规定。	类型由 PICTURE 子句规定。
2. 内务处理(打开与关闭文件,检验标号,为了执行而装入数据)	由编译程序提供。	SELECT, OPEN 和 CLOSE 文件, INITIATE 和 TERMINATE 报告。
3. 输入	READ 语句。	READ 和 ACCEPT 语句。
4. 处理	这一形式的赋值语句: 变量 = 表达式。 算术语句。 循环处理。 DO 循环按规定好的次数执行某些过程。 没有提出。	MOVE(非破坏性)语句: MOVE 标识符-1 TO 标识符-2。 ADD, SUBTRACT, MULTIPLY, DIVIDE 和 COMPUTE 动词。 PERFORM 语句。
	机器内部生成的变量不需要在格式语句中定义。	字符处置动词: EXAMINE, INSPECT, STRING 和 UNSTRING 等。 对于每一个在过程部分中使用的由程序员提供的数名,必须在数据部分中给它一个 PICTURE 子句。
5. 数据结构	变量名在赋值语句, DATA 语句和 TYPE 语句中的定义。双精度型、复数型与逻辑型变量使 FORTRAN 容易适应统计和科学问题。	文件,记录,组合项和基本项的逻辑结构接近于商业实际。
6. 复杂的输入与输出运算(要求数据以这样的方式进行组织,以尽量减少查找和检索的时间)	在规模和能力上均比较差。	具有强有力的数据管理和文件存取的能力。
7. 输出	WRITE 语句。	WRITE 和 DISPLAY 语句。
8. 其他	可以应用数学子程序以简化程序的准备工作。	更加容易适应成批控制的商业应用问题。

图 1.6 FORTRAN 与 COBOL 比较表