微型计算机

COBOL

程序设计语言

上海市计算技术研究所

1981·1

编　者　前　言

COBOL语言（Common Business Oriened Language）是一种通用的高级程序设计语言。它是明确为供企业数据处理之用而制定的。它产生于五十年代末期，经多年发展、完善，已为世界上绝大多数计算机制造厂和用户接受。目前应用最广的是美国国家标准协会（ANSI）1974年发表的标准文本，称为ANS－COBOL，它也是国际标准化组织（ISO）1978年推荐的国际标准COBOL的蓝本。

为了实现与使用方便，标准的COBOL由十二种功能模块组成，每种功能模块又分为一至二级，低级总是高级的一个子集。最小的COBOL至少包含一级核心、表处理和顺序存取。

核心	表处理	顺序存取	相对存取	索引存取	报表打印	排序和合并	分块	库	调试	内部通讯	通讯
二级	二级	二级	二级	二级	一级	二级	二级	二级	二级	二级	二级
			一级	一级		一级	一级	一级	一级	一级	一级
一级	一级	一级	0级	0级	0级	0级	0级	0级	0级	0级	0级

表一　COBOL语言的功能模块和级

COBOL语言得到如此广泛应用的主要原因，是因为它具有如下几个特点：

Ⅰ）　COBOL语言面向文件，以文件作为主要处理对象，而文件是绝大多数事务数据处理系统的基础。

Ⅱ）　COBOL语言高度独立于计算机，为某一台计算机设计的COBOL程序，只需作少量的关于设备描述的修改，即可适用于另一型号的计算机。

Ⅲ）　COBOL程序的描述接近于自然英语，便于阅读，尤其容易为使用和懂得英语的人们接受。

COBOL语言的标准文本的实施，依赖于具体型号的计算机上的COBOL编译程序，每个COBOL编译程序又必须在具体的操作系统管理下进行。本书介绍的CROMEMCO COBOL是ANS—COBOL的一个子集。它是在CROMEMCO微处理机系统上实现的一个COBOL语言文本。包含除了报表打印，排序合并，分块和通讯之外的八个功能模块的全部一级功能和部分二级功能。

CROMEMCO COBOL运行的物理基础是CROMEMCO微型计算机硬件系统，它以Z—80A组成的CPU为核心，配置键盘显示终端，点阵式打印机和软盘存贮器作为基本外围设备。磁盘操作系统CDOS（CROMEMCO DISK OPERATING SYSTEM）具有控制内外存交换，管理磁盘文件，解释用户命令等功能，COBOL源程序的生成、编译、连接和执行都在CDOS的控制下进行。（有关CDOS的详细介绍，请参阅“Cromemco软件资料（一）（清华大学编）”。

近年来，微型计算机系统及其应用发展极为迅速，在国内也越来越引起重视，用户越来越多。为充分发挥微型计算机的功能，进一步推广它的应用，特编制这本简单的资料，供打算在微型计算机上应用COBOL语言进行数据处理的用户参考。本书编写力求完整。既可供初学者作为教材，又可供使用者作为手册，这是编者的初衷，为尽可能减少篇幅，关于语法说明的实例均未列入，若用之作为教材，需由教师加以补充。由于水平有限，又是初次尝试，时间仓促，错误及不当之处在所难免，切盼读者不吝指教，深表感谢。

上海计算技术研究所

裴　广　生

一九八〇年九月

第一章　COBOL的基本概念

1.1　文件的特性

COBOL语言“面向文件”，用这种语言编制的程序可以用来描述文件、建立文件、阅读文件、维护文件和对文件中的数据进行处理等。在详细描述COBOL语言之前，简单介绍文件的一些属性。

1.1.1　数据的层次结构

在企事业的实际业务活动进程中，伴随着产生大量的信息流动。记录这些信息怎样随着实际业务活动而产生、发展和终结，就是所谓“事务性数据处理”。录制的信息被称为“事务数据”或“数据”。手工方式的事务处理系统中，往往根据业务的需要，将不同的数据，按照它们的相互关系组合成各种类型的书面凭证（传票、单据、报表等）并将同类型的凭证归并成档案或帐册。

跟书面的帐册、档案等相对应，我们将存放在计算机外部存贮介质（磁盘、打印机等）上，按一定方式组织起来的数据记录的集合称之为“文件”。所谓“记录”，类似于凭证，它是构成文件的单位，实际上，通过逐个处理文件中的记录实现对整个文件的处理，记录是文件处理过程中一次处理的对象。另一方面，记录又是若干逻辑上相关的数据项的合成体，记录中的各个数据项按事先指定的次序前后相接连续排列，记录中的数据项若在逻辑上还可以细分，则被称为组项，逻辑上不能再细分的具有独立含义的数据项，被称为初等项。初等项在形式上由若干没有独立意义的字符组成。

数据之间的这种内在的逻辑关系可以用层次结构来反映，同一层次的数据之间存在平列关系，不同层次的数据之间存在从属关系。COBOL语言用不同的层号来表示数据的不同的层次，并且能在几个层次的任一层次上引索和使用数据。

1.1.2　文件与设备

文件必须建立在计算机外部存贮介质上，对于微处理机系统而言存在磁盘文件和打印文件两种类型的文件。

其数据通过点阵式打印机按照一定格式直接打印在纸上的文件称

为打印文件。打印文件的数据逐行顺序输出，每个打印行是一个记录。一个打印行可印刷120个字符，每个字符跟计算机中一个八位二进制数相对应，因而打印文件的记录长度固定为120字节。

磁盘文件中的数据存贮在软磁盘上。软磁盘是塑料的圆盘，表面涂有磁性材料，盘面上由磁点构成一个个同心圆环，称为磁道，数据录制在磁道上，从外沿向中心对磁道进行编号。每个磁道又分为若干区段。CROMEMCO系统可使用5英寸或8英寸二种盘，前者每盘面有40个磁道（编号0－39），每磁道分18个区段，后者每盘面有77个磁道（编号0－76），每磁道分26个区段。每区段存贮空间为128个字节。5英寸的单面磁盘容量为90K字节，双面磁盘容量为181K字节，8英寸的单面磁盘容量为256K字节，双面磁盘容量为512K字节。每个磁盘上由系统软件占用约8.5K字节的位置，因而实际提供用户使用的文件区大小，要从磁盘容量中减去系统占用区容量。

磁盘文件的记录长度由程序员根据实际业务的需要设定。CDOS将提供用户使用的磁盘文件区分簇，每簇包含1024字节，在同一簇上只能存放同一文件的数据，簇中未被该文件占用的存贮单元，不能存放其它文件数据，因此，为了充分利用磁盘空间，用户必须合理安排磁盘文件的记录长度。

1.1.3 文件的组织方式

文件的组织方式指文件中记录的排列方式。它在文件建立时被确定，在文件整个使用时期内，它的组织方式不变。文件的组织方式大致有如下三种：

Ⅰ） 顺序组织指文件中的记录按连续的顺序排列，文件中记录被读出的顺序与文件中记录排列的顺序以及建立文件时记录写入的顺序，三者一致。打印机文件就是典型的顺序组织文件。磁盘文件也可以进行顺序组织。

Ⅱ） 相对组织指文件中的每个记录对应一个相对记录号，根据相对记录号确定记录在文件中的位置。只有磁盘文件可按相对方式组织。

Ⅲ）索引组织指文件的每个记录中存在一个称为记录键的数据项，在文件中记录按记录键的值的递增次序排列，同时，根据键值建立控制索引表，以对每个记录的实际存贮区域定位。只有磁盘文件可按索引方式组织。

1．1．4　文件的存取方式

从文件向内存读取数据是输入操作，这时，文件被称为输入文件，从内存向文件存入数据是输出操作，这时文件被称为输出文件。在同一处理过程中既作输入又作输出的文件被称为I—O文件。文件存取的几种基本方式及其与文件组织方式的关系是：

Ⅰ）顺序存取：按记录在文件中的排列次序进行存取。对顺序组织的文件，只能顺序存取，这时，每次读或写的记录总是上次读或写的记录的下一个记录。对相对组织文件，按相对记录号的递增次序顺序存取。对索引组织的文件，按记录键的值的递增次序顺序存取。

顺序存取方式处理简单，就处理整个文件而言，速度较快，但处理灵活性差。

Ⅱ）随机存取：按程序员指定的次序存取文件中的记录，一般而言，这种次序与文件中记录排列次序无关，只有磁盘文件可按随机方式存取。程序员通过指定相对记录号或记录键的值，在相对组织文件或索引组织文件中随机地存取所需的记录。

Ⅲ）动态存取是指对同一文件的同一处理过程（从打开文件开始到关闭文件结束）中，交替地或任意地采用顺序存取和随机存取两种方法。

1．1．5　文件缓冲区

对于每个文件，计算机内存贮器中有一个存贮区域与之对应，称为文件缓冲区，它的大小跟文件的记录的长度一致。它是文件中每个记录公用的存贮区域。每次从文件读取的数据进入该文件的缓冲区，每次向文件写入的数据亦来自它的缓冲区。除了输入／输出操作之外，对文件的记录内的数据项的处理，实质上是对文件缓冲区内与数据项相应的存贮区域中寄存的数据进行处理。

1．2　COBOL 语言概述

1．2．1　COBOL 字符集

计算机内部用二进制形式表示信息。一般用八位二进制数表示一个字符，Cromemco 微机系统使用国际通用的ASCII 字符集。COBOL 语言字符集是ASCII字符集的一个子集，它包含下列五十二个代码：

字　符	十六进制代码	字　符	十六进制代码	字　符	十六进制代码	字　符	十六进制代码
A	41	N	4E	0	30	'	27
B	42	O	4F	1	31	(	28
C	43	P	50	2	32	)	29
D	44	Q	51	3	33	*	2A
E	45	R	52	4	34	+	2B
F	46	S	53	5	35	,	2C
G	47	T	54	6	36	-	2D
H	48	U	55	7	37	.	2E
I	49	V	56	8	38	/	2F
J	4A	W	57	9	39	;	3B
K	4B	X	58	空格	20	<	3C
L	4C	Y	59	"	22	=	3D
M	4D	Z	5A	$	24	>	3E

除了在引号内的非数值常字及注解行和注解项中可以出现ASCII字符集中的任意字符之外，COBOL语言的所有语法单位都只能由以上52个字符构成。

一系列连续的字符构成字符串，字符串之间用分隔符定界。分隔符包括标点符，算术运算符和关系运算符。

标点符是：␣ ， ； . " ' ( )

算术运算符是：＋ － * ／ * *

关系运算符是：< = >

COBOL 源程序的正文无非是一些字符串和分隔符的组合。

1．2．2　COBOL 字

一个COBOL 字是一个不超过３０个字符的字符串。字的组成规则如下：

Ⅰ）　字仅由字母A～Z，数字0～9和连字符“－”等３７个字符的任何组合构成。且必须以字母开头。

Ⅱ）　一个或多个连字符可出现在字的内部，但不能作为字的结尾。相连的连字符可只保留一个。

Ⅲ）　字的两端由分隔符定界。

COBOL 字按其性质可分为保留字和程序员定义字两种。

1．2．2．1　保留字

一些英语单词和缩写，在COBOL 语言中具有预先规定的语法意义，称之为保留字。使用保留字时必须书写正确。（保留字表见附录）

1．2．2．2．程序员定义字

由程序员定义的COBOL 字，亦称为名字。按用途可分为数据名，条件名、过程名、记忆名及层号等。

Ⅰ）　数据名和层号

数据名用于命名一个数据项。实际上数据名亦是赋以该数据项在它所属的文件的缓冲区中所占用的存贮区域的名称。这些区域都用其相应的数据名来引索。

在COBOL 语言中，定义一个数据名时，在它前面要赋以层号，以此反映数据的层次结构，记录的层号固定为０１，从属于记录的数据项的层号可以是02～49之间的数字，在同一记录中数据项的层号不必连续，平列的数据项必须有相同的层号，下属的数据项必须有较高的层号。

存放数据处理中间结果的数据存贮单元和一些不属于任何文件的独立数据被安置在工作存贮节中，也可用数据名来命名其中不同的存贮区域，并用层号来反映这类数据的逻辑上的并列和从属关系。特别地，对跟其他数据没有逻辑关系的独立数据项，赋以专用的层号７７。

跟数据名相类似，用文件名来为程序中使用的每个文件命名，文

件名的定义应该唯一。文件亦是数据的一个层次，它的层号是专用的保留字FD（亦称层指示符）。

Ⅱ） 条件名

数据名类似于代数中的变量，它是一个标识数据项的符号，可以有许多可能的值。条件名用来标识相应的数据名的值的状态，即用一个COBOL字来指称相应数据名的一个值或一组值或一个值的范围。定义条件名时，给它赋以专用的层号88。

Ⅲ） 过程名

说明程序运行时要执行的动作的是语句，若干语句的集合组成过程体，过程体按其层次的不同可分为节和段，节由段组成，段亦可单独存在。命名过程体的字称为过程名，因而段名和节名都是过程名。特别地，过程名可以仅由数字组成。

Ⅳ） 记忆名

对于所选择的用于接受和显示少量数据的设备，可用程序员定义字命名，称为记忆名。

1.2.3 常字

常字是一个字符串，它是数据项的另一种表达方式，在程序中直接用来表示一个常数，在程序运行过程中始终不变。常字包括非数值常字和数值常字两种，均由程序员定义。

1.2.3.1 非数值常字

两端由引号（单引号或双引号）定界的字符串是非数值常字。字符串内可以包括除上述定界符之外的任意ASCII字符。常字的长度是定界符之间的字符个数，最大允许长度为120，最短长度为1。非数值常字的值是组成它的字符串的字面值。例如，“12.5”表示四个字符的字符串，可以用它的十六进制代码形式“3132E35”跟其他非数值数据项进行比较，但它不表示数值0.125×10^{2}，亦不能参加算术运算。

1.2.3.2 数值常字

数值常字由数字0～9及符号+、－和小数点等13个字符按如下规则组成：

Ⅰ） 至少有一个数字，至多有18个数字。

Ⅱ） 至多只能有一个符号，它若出现，则必须是字符串的第一个字符。符号若不使用，就假设它是正的。

Ⅲ） 至多只能含一个小数点，它可以出现在字符串中除最右位置外的任何位置上。若不含小数点，就认为该常字是整数。

数值常字的值就是组成它的字符所表现的代数值。

1．2．4 象征常数

具有固定名称的常字叫象征常数，它们是一些保留字，不以引号作定界符。它们的名称正好指出了它们的值。在语法格式中，凡是可以使用非数值常字的地方都可以使用象征常数，凡是允许数值常字出现的地方都可以使用ZERO 这个象征常数。编译程序将源程序中出现的象征常数用一定的常数值来取代，见表二中的说明部分。

象征常数	说明
ZERO．ZEROS ZEROES	根据上下文，产生数值零或者一个（或多个）字符“0”。
SPACE SPACES	产生一个或多个空格
LOW—VALUE LOW—VALUES	产生一个或多个用十六进制代码“00”表示的字符
HIGH—VALUE HIGH—VALUES	产生一个或多个用十六进制代码“FF”表示的字符
QUOTE QUOTES	产生一个引号，但不能用它来作非数值常字的定界符
ALL＜常字＞	表示组成该常字的字符串的一次或多次重复。这里常字可以为非数值常字及除ALL之外的所有象征常数，不能为数值常字

表二 象征常数

象征常数表示的字符长度，跟语句中与它相关的项所要求的长度一致。当象征常数与其它数据项无关时，它只表示一个字符。在任何情况下使用QUOTE只产生一个引号。

1．2．5 COBOL 源程序的逻辑结构

用COBOL 语言编写的程序叫COBOL 源程序，每个COBOL 源程序由四个部分组成，它们的出现次序和作用是：

标识部分	给程序命名
环境部分	指出程序运行时使用的计算机设备及特点。
数据部分	定义被处理的数据的名称和特征
过程部分	包含程序执行时直接处理数据的语句

下列编码梗概描述了COBOL 程序元素的结构和顺序：

```
IDENTIFICATION  DIVISION
PROGRAM－ID．程序名
〔AUTHOR ．          〔关于作者名的注释项〕〕
〔INSTALLATION ．〔关于程序所属系统或应用范围的注释
                    项。〕〕
〔DATE－WRITTEN ．〔关于程序书写日期的注释项〕〕
〔DATE－COMPILED.〔关于程序编译日期的注释项〕〕
〔SECURITY ．        〔关于程序保密限制的注释项。〕〕
ENVIRONMENT DIVISION ．
〔CONFIGURATION SECTION ．
〔SOURCE－COMPUTER ．关于源计算机的描述体〕
〔DBJECT－COMPUTER ．关于目标计算机的注释项〕
〔SPECIAL－NAMES ．    描述体〕〕
〔INPUT－OUTPUT SECTION.
FILE－CONTROL ．      描述体。
〔I－O－CONTROL ．    描述体。〕〕
DATA DIVISION ．
〔FILE SECTION.
〔文件描述体
```

记录描述体……〕……〕

〔 WORKING — STORAGE SECTION.

〔 数据项描述体……〕……〕

〔 LINKAGE SECTION.

〔 数据项描述体……〕……〕

PROCEDURE DIVISION 〔 USING 标识符－1…〕

〔 DECLARATIVES.

｛节名 SECTION.USE 句子。

〔段名。〔句子〕……〕……｝ ……

END DECLARATIVES.〕

｛节名 SECTION .

〔段名。〔句子〕……〕……｝……

其中注释项只作为程序员的注解，没有实际的语法意义，以后不再述及。

描述体由若干相继的子句构成，子句是一种类似于英语中主表结构的描述性语法单元。

句子由若干相继的语句构成，以句号结束。语句由动词开头，后跟相应的操作数和短语。

段由段标题开头，段标题由段名和紧跟着的句号及空格构成，段内包含描述体或句子。一个段直到遇到下一个段名、节名或部分标题时结束。

节由节标题和后随的0个，1个或多个相继的段组成。节标题由节名跟空格和字 SECTION 及句号和空格构成。在过程部分中如果使用了节，则所有的段必须在节中，一个节直到遇到另一个节标题或部分标题或程序终止时结束。

部分标题由部分名跟空格和字DIVISION 及句号和空格构成。

1．2．6 COBOL 源程序书写格式

COBOL 源程序书写在标准的编码纸上，每行由80 列构成，分为五个区，第1～6列是序号区，第7列是指示区，第8～11 列是A区，第12～72列是B区，第73～80 列是注释区。

序号区中用数字标识源程序中的行，可以省略，若使用，则需按照序号数的升序排列（不必连续）。

指示区中可出现字符为*，／，－，D，它们的作用：

*—— 表示该行为注解行，这样的行只出现在源程序表中，编译程序对它不作其它处理。

／—— 表示该行为注解行；但编译程序在打印源程序表时，将这样的行打在新的一页的顶部。

－—— 表示该行为继续行。当一个字（例如数据名）或数值常字分写在两行时，连字符表示该行B区的第一个非空格字符逻辑上紧跟在上一行B区最后一个非空格字符之后。当一个非数值常字分写在两行时，继续行B区的任何位置上可以出现一个引号，表示引号之后的字符紧跟在上一行的第72列字符之后。

D—— 表示该行是调试行。

A区和B区用于写源程序的正文。部分标题，节标题都必须单独占一行。部分名、节名、段名、层指示符FD，层号01和77都必须从第八列开始书写，其他层号可以出现在A区和B区的任意位置上。程序中的所有其他元素都必须写在B区中。

注解区的作用跟注解行的作用类似。

1．2．7 分隔符使用规则

Ⅰ） 两个相邻字符串之间至少必须用一个空格分开，除了在非数值常字中外，凡是可用一个空格的地方，均可用多个空格。

Ⅱ） 逗号可用来分隔语句中的一串操作数，逗号与分号均可用来分隔子句或语句，在这些场合，逗号与分号都可被空格代替。句号必须紧跟在每个句子或描述过体之后。当逗号、分号，句号用作分隔符时，必须紧跟在非空格字符之后，其后必须紧跟空格。

Ⅲ） 算术运算符和关系运算符的前后，左括号之前，右括号之后都必须跟空格相连。单目运算符（正负号）出现在数据名之前时，其后必须紧跟空格，出现在数值常字之前时，其后不能连接空格。单目运算符之前可与空格或左括号直接相连。

1．2．8 语言描述方式说明

Ⅰ） 语法格式中所有用英文大写字母给出的都是保留字。下面加横线的是必写的关键字，否则是可写可不写的任选字。

Ⅱ） 语法格式中出现的各种分隔符，标识它们在程序中应出现的位置，它们是不可省略的。

Ⅲ） 用中文字书写的字表示要由程序员加入的信息。同一语法格式中出现具有相同作用的不同项目时，用中文字后跟一个连字符和一个数字或字母（例如标识符－m），以资区别。语法元素的含义不变。

Ⅳ） 格式中的〔 〕、{ } 和……是描述记号，不能出现在源程序中，它们的用途如下：

〔 〕—— 表示其中的语法元素是任选的。

{ } —— 其中必有两个以上的语法元素说明项，使用时，必须选用其中的一个。

…… —— 一般跟在〔 〕或{ }之后，表示括号中出现的整个说明项可以出现多次。

1．2．9 名字的限定

COBOL 语言允许程序中出现的数据名，条件名或段名不唯一，这时，可以通过名字后跟限定符的方式进行受限，使过程部分中可以唯一地引用这些名字。限定符以 OF 或 IN 引导，数据名可用其组项的名字作为限定符，条件名可用其条件变量或条件变量的组项的名字作限定符，段名可用节名作限定符。段名的限定符至多1个，数据名和条件名的限定符至多5个。文件名、记忆名和节名在程序中必须唯一。

带限定符的名字只在过程部分出现。

1．2．10 拷贝语句

拷贝语句的格式是

COPY 文本标记。

该语句将在磁盘中保存的以文本标记标识的标准COBOL 编码实体结合进源程序，插在 COPY 语句出现的地方，一起进行编译，这样

就可以利用已设计好的标准程序部分，提高编制程序的效率.

COPY 语句可以出现在程序中的任何地方，但在被拷贝的程序实体身之内不可出现. 拷贝语句在源程序中必须单独占一行，以句号结束.

第二章　标识部分和环境部分

2.1　标识部分

标识部分的一般格式和在源程中位置如1.2.5.节所述。其中PROGRAM-ID段是必须的。用来提供程序标识。段中的程序名是一个程序员定义字。编译程序识别程序名的最前面六个字符。

2.2　环境部分

环境部分用来说明一个程序编译和运行时所依赖的主机和外部设备及其特性。

2.2.1　配置节(CONFIGURATION SECTION)

配置节包括三个具有固定名称的段：

I)　源计算机段。其格式为

```
SOURCE-COMPUTER 。计算机名 [WITH DEBUGGING MODE]
```

当使用DEBUGGING MODE短语时，程序中所有的调试行跟源程序其他部分一起被编译，否则，这些行被看作注解行。本段的其它部分仅作注释，供程序用户参考。

II)　目标计算机段，其格式为

```
OBJECT-COMPUTER [计算机名] [MEMORY SIZE 整数 {WORDS / CHARACTERS / MODULES}]
    [PROGRAM COLLATING SEQUENCE IS ASCII]
```

本段仅作注释。供程序用户参考。

本段仅作注释，供程序用户参考。

IV)　专用名段。其格式为：

SPECIAL-NAMES[PRINTER IS 记忆名]ASCII IS{STANDARD-1 / NATIVE}

[CURRENCY SIGN IS 常字][DECIMAL-POINT IS COMMA]

用法说明：

1） 若使用PRINTER 子句，则对打印机定义一个记忆名，它被用在过程部分中 DISPLAY 语句的UPON短语内，表示在打印机上显示数据。这样，如果使用的显示设备变动时，只需对本子句作修改，而不必涉及过程部分中的语句。

2） 用户可用CURRENCY 子句定义一个币制符号，这里的常字是单个字符的非数值常字，不可为引号或数字，亦不可是构成形象字符串的形象字符。当不用本子句时，以美元符号$作为币制符号。

3） 使用DECIMAL-POINT 子句导致在数值项中，逗号和小数点的功能互换。

4） ASCII 子句实际上仅起注释的作用。

2．2．2 输入一输出节（INPT--OUTPUT SECTION）

本节包含两个具有固定名字的段

2．2．2．1 文件控制段

本段说明本程序运用时用到的所有文件及其物理特性。

其格式为：