# 微型机软件

COBOL—80

## 参考资料

上海电子计算机厂翻印

# 前　言

鉴于ＢＣＭ微型计算机制造厂和用户的迫切需要，我们翻译了本资料，以供参考。

由于时间匆忙和我们的水平有限，本资料在翻译和印刷等方面一定会有不少错误，请读者谅解，并请及时批评指正。

清华大学经管系实验室

1982·1

## 微型机软件COBOL—80资料

微型机软件COBOL—80及其附带软件包括以下资料：

1. COBOL—80用户指南

叙述关于运行COBOL—80，写COBOL程序和用你的硬件运行程序的全部方法。

2. COBOL—80参考手册

提供了COBOL—80的语句，语法和结构等方面的描述。

3. MICROSOFT实用软件手册

叙述MACRO—80汇编程序，LINK—80连接程序和带有COBOL—80编译程序的LIB—80程序库管理程序的应用。

# （一）

# COBOL—80用户指南

# COBOL—80用户指南目录

# COBOL—80用户指南

## 第一节 综述

### 1·1 引言

这本COBOL—80用户指南目的是给你实际的有关编制COBOL—80程序并且用你的计算机设备运行的信息。成功地用COBOL—80的一切必要步骤——编辑、输入、执行等等——都在下面详细讨论。

本指南所给的例子和文件名是基于COBOL—80的CP/M版本，如果你用另外一种操作系统，命令和文件名的格式将略微有些不同。看附录D，用你的操作系统COBOL如何描述。

### 1·2 你的磁盘分配

你从Microsoft得到的磁盘有下列文件：

COBOL编译

COBOL.COM

COBOL1.OVR

COBOL2.OVR

COBOL3.OVR

COBOL4.OVR

(runtime system)

COBL1B. REL——运行时程序库

(the runtime library)

CRT 驱动器 ——名字以CD开头的文件

源文件—— C D—— MAC

目标文件—— CD——. REL. CRTDRV REL

实用软件

L80. COM—— Microsoft 联接装配程序

L1B. COM—— Microsoft 库管理

M80. COM—— Microsoft 宏指令汇编程序

CREF80. COM—— Microsoft 汇编交叉参考程序

各样文件

SQUARO. COB

CRTEST. COB

SEQCVT. COM

COPCOB. SUB

### 1·2·1 COBOL 编译

编译包括一个主程序和四个复盖程序，这五部分相当于五个编译阶段。主程序常驻内存，并且控制每一个阶段到下一阶段的转移。主程序的复盖程序部分编译环境部和标识部。OVErlay1 被引入编译数据部，过程部由 Orerlay2 编译，这三部分组成编译的第一次扫视。它们的功能是建立一个程序的中间文本，文本存贮在文件名为 STEXT. IMT 的现行磁盘上。OVerlay3 引导中间文件，并且建立目标码，最后 OVerlay4 分配文件控制块，并且检查一定的错误条件。中间文件随后被删除。

### 1·2·2 运行时系统

运行时程序库包括一组子程序，来解释由编译产生的你的程序

的目的码。当你完成输入步骤时（看手册第三章）这些子程序将被包含在你的目的程序中。并非所有的程序需要一切库程序。装载程序器将检查这个库，并且自动包括你所需要的部分。提供的CRT驱动程序能使你把你的系统成形，并用你有的CRT终端打字。你将需要选择适当的驱动器（看本手册附录A），一旦你选定了，驱动器将被自动地包含在你用联接装载程序输入的每个程序中。驱动程序提供光标定位和其它功能，以支持ACCEPT和DISPLAY语句的相互作用。

1·2·3 实用软件

Microsoft联接装载程序被用来联接COBOL目标程序和运行时系统（看本手册第三章），其它的应用为您提供方便。这些程序中的每一个在Microsoft用户软件手册中提供资料。

1·2·4 各种各样文件

SQUARO.COB是COBOL源程序，这个程序计算你给的数的平方根 它用来检验你的编译和运行时系统的工作版本。

CRTEST.COB是COBOL源程序，它用来试验交互CRT驱动程序的功能（看附录A）。

SEQCVT.COM是一个专门的应用程序，它使COB文件从行顺序格式转到顺序格式。当版本3·0被再版（release），COBOL 80顺序文件格式被改变。顺序组织文件由这些版本建立是所知道的行顺序格式。

COPCOB.SUB是一个命令文件，他把你的分配磁盘转贝到第二块磁盘。它是作为一种便利提供的。

1·3 起动

当你得到你的磁盘后应该做的第一件事情是进行拷贝。用它并存在原始磁盘上作后备。这可以利用所提供的COPCOB命文件来做，或用一些你可能有的其它磁盘拷贝的便利条件。

做完那些，你应该用编译，装载和执行试验程序SQUARO·COB来证明你的编译和运行时系统的拷贝是正确的。做完这些参考下面1·4节给的例子。最后，你想用交叉式ACCEPT和DISPLAY以方便你的COBOL程序，你必须选择CRT驱动程序，并把它在你的运行时系统中成形。这个过程只需要做一次，此后，你选择的驱动程序将自动地被包含在你的每个目的程序中，看本手册附录A的所有指令。

## 1·4 程序开发步骤

要执行COBOL程序的准备工作包括三个基本步骤：

①用文本编辑建立源文件

②用COBOL编译程序进行编译

③用联接装载程序进行装载。

源程序是一个文件，是由ASCII码文本行组成，由回车换行结束。你也能用Microsoft的EDIT—80或任何别的用7—bit、ASCII码的编辑器来建立。行号要放在每行的1～6列，且可以是8—bit ASCII码，除TAB和回车外，编译程序忽略7列前的这些字符。编译程序假设TAB停止的列是7、17、25、33、41、49、57、65和73列。在73列后，所有的字符被忽略，直到遇到回车。如果你用EDIT—80，你自动地在每个插入行的第7列开始打印。

建立源文件后，下一步就是编译它，这通过打印一个执行COBOL

建立源文件后，下一步就是编译它，这通过打印一个执行COBOL编译程序和提供你的源程序名的命令来做到的。在CP/M操作系统下，你必须在包括COBOL编译的磁盘上请求联机，因为编译复盖程序总是要从现行磁盘上读的。下列例子说明用来编译试验程序SQUARO的一个命令，它包含在你的磁盘分配上，假定磁盘在驱动器A上。

A > COBOLSQUARO。REL，TTY：=SQUARO。COB

这个命令将编译SQUARO。COB，在文件名为SQUARO。REL中放浮动目的码，并且在你的终端上打印清单。这命令的较短的表示法，不写由编译假定的文件扩展名也可以：

A > COBOLSQUARO，TTY：=SQUARO

最短的表示法，应用一个编译开关，使得产生一个目的文件，而省略了文件名SQUARO。REL

A > COBOL，TTY：=SQUARO/R

这三个命令实际上都产生相同的效果。第二章给出命令行句法的说明。源程序一旦被编译，在执行以前的最后一步是用联接装载程序L80装载的。这一步把你的浮动目的程序转换成绝对(absolute)文本，并且把它与COBOL—80运行时系统结合起来，这绝对文本在内存中被建立。它随后可以存在磁盘上，直接执行，或者又存贮又执行。

下面是一个装载SQUARO并且执行它，而不存绝对形式的命令：

A > L80 SQUARO/G

L80假设被装载的文件SQUARO的扩展名，REL；一旦SQUARO被执行，你不用装载命令就不能再执行它，因为绝对文本没存。在一个磁盘文件上存绝对文本而不立即执行它，打印：

A > L80 SQUARO/V , SQUARO/E

随后执行这个程序，只要简单地打印：

A > SQUARO

因为绝对文本被存了，在任何时候不需要执行装载步骤都可以执行。结合2例，绝对文本被存随后直接执行，打印：

A > L80 SQUARO/И, SQUARO/G

L80命令的描述参考本手册第三章，和Microsoft实用软件手册。

## 第二节　编译COBOL程序

### 2·1 COBOL—80命令行句法

COBOL 80编译程序当作输入读你的COBOL源程序文件，并且产生一个清单和你的程序的浮动目的模块。命令请求COBOL编译程序，并且告诉他所用的三个文件的名字。行句法是打印COBOL跟一个空格，跟一个如下描述的命令串。如果对，编译开始。如果不对，打印信号"？ Command Error"，（？"命令错"），跟一个*号提醒，随后等待另一个命令串，当编译完成，COBOL 80总是回到操作系统。

COBOL—80命令串的格式是：

目标文件，清单文件=源文件

分隔符是逗号和等号，不允许空格，在格式中所用的项是：

目标文件：所写的目标程序的文件名

清单文件：所写的清单程序的文件名

源文件：　COBOL程序源文件名

每个文件可以是磁盘文件的名或系统装置的名，文件名的描述，依

于你的操作系统，对CP/M，文件描述有这样的形式：

装置：文件名，扩展名

这儿分隔符是冒号和句号；项的意思是一个磁盘驱动器，终端、行打印或操作系统提供的别的装置。如果这装置是磁盘，必须给磁盘名，如果不是磁盘，装置名本身是文件描述，COBOL80认识下列符号装置名字：

TTY：终端控制台

LST：系统打印机

RDR：高速读数机

文件名

磁盘上的文件名，如果文件名是专用的，没有装置，那么现行的磁盘被假定作装置。

扩展名

给的文件的扩展名，如果没说明，下面的省略被假定。

.COB 源程序文件

.PRN 清单文件

.REL 目标程序文件

在命令串中，目标文件，清单文件或二者都可以被省略，如果既不需要清单文件，也不需要目标文件，COBOL—80将检查错误。并且打印在控制台上。如果等号的左边什么也不打印，一个目的文件被写在相同的设备上，且用和源文件相同的文件名。

例：

| 命令串 | 功能 |
| --- | --- |
| =PAYROLL | 编译源程序PAYROLL.COB并且只产生一个错误，被显示在控制台上。 |

| | |
|---|---|
| =PAYROLL | 编译 PAYROLL.COB 并且放目的程序到 PAYROLL.REL 不产生清单。 |
| ,TTY:=PAYROLL | 编译 PAYROLL.COB 源程序，并且放程序清单到终端显示器，不产生目标程序。 |
| PAYROLL.LST:=PAYROLL | 编译 PAYROLL.COB 源程序，放清单到打印机，并且放目标程序到 PAYROLL.REL。 |
| PAYOBJ=B:PAYROLL | 在磁盘 B 上编译 PAYROLL.COB 并且放目的程序到 PAYROLL.REL，不产生清单。 |
| PAYROLL,PAYROLL=PAYROLL | 编译 PAYROLL.COB 放清单文件到 PAYROLL.PRN，并且放目标程序到 PAYROLL.REL。 |

## 2·2 编译开关

命令串可以通过加上一个或多个开关来说明，开关象如下描述的影响编译过程。加上一个开关，打印一个/跟一个字符的开关名。

| 开关 | 功能 |
|---|---|
| R | 使编译程序产生一个目的文件，这个简写标志使编译程序写一个目的文件在相同的磁盘上，并且用和源文件相 |

同的文件名，但是省略目的文件的扩展名。

L　使编译程序产生一个清单文件，和用／R一样，这个符号使编译写一个清单文件在相同的磁盘上，并和源程序用相同的文件名，但是省略清单文件的扩展名。

P　每个／P分配额外的100字节存贮空间给编译程序用。在编译中如果存贮溢出错误发生用／P（看2·3节），否则／P是不需要的。

用开关的命令串例子：

| 命令串 | 是等效于 |
|---|---|
| ,=PAYROLL/R | PAYHOLL=PAYROLL或PAYROLL |
| ,=B: PAYROLL/L | ,B: PAYPOLL=B: PAYROLL |
| ,=B: PAYROLL/R/L | B: PAYROLL,B: PAYROLL<br>=B: PAYROLL |
| =PAYROLL/L/P | PAYROLL,PAYROLL=PAYROLL/P |

## 2·3 输出清单和错误信息

由COBOL—80输出的列表文件是带页头和错误信息的源文件的一行接一行的报表（account），页头行从页顶被打印3行，跟二个空行。每个被列表的原始行是以一个顺序的4位十进制数开始。这在最后被错误信息用来参考错在哪行。在错误发生后，也被运行时系统用来说明什么语句引起运行错误。

在编译中可能产生二类错误。低级标记（Lou level flag）直接描述在清单原行的下面，当简单的违反句法发生时，在每种情况下。象下面列的文件，校正的作用被采取，而且编译继续。如果

低级错误发生，一个高级的特征在清单尾被产生。参考行标号所附的低级错误。所以在末尾给的错误信息包含了两类错误。

| 标记 | 标志原因 | 编译程序的校正作用 |
| --- | --- | --- |
| "QLIT"? | 错误地引用文字 | |
| | 1.零长度 | 忽略并继续 |
| | 2.错误的顺序 | 假定可接受 |
| | 3.过早的结束文件（end—of—file）（在结束定义符前） | 假定程序结束 |
| LENGTH | 引用的文字长度超过120字符，或数字超过18位，或"字"（标识或名字）超过30字符。 | 多余的字符被忽略 |
| CHRCTR | 禁用字符 | 忽略并继续 |
| PUNCT? | 错误的标点（即逗号没跟空格） | 假定可接受 |
| BADWORD | 现行字是畸形的，比如：以"—"结束，或一个十进制数中有多个小数点。 | 忽略并继续 |
| SEQ非 | 错误的顺号（包括这种情况out—of—order Sequence | 接受和继续 |

| | | |
|---|---|---|
| | order Sequence number ) | |
| NAME? | 名字没以A—Z开始 | 接受并继续 |
| PIC = X | 错误的PICTURE | 假定是PIC X |
| COL 7? | 一个错误的字符出现在原行的第7列，第7列只允许*、—/、D | 假定第7列是空格 |
| AREA A? | A区8～12列在一个继续行中不是空格 | 忽略A区内容（假定是空格） |

高级特征信息由二或三部分组成：

1. 有关的原行号——4位跟一个冒号（：）。

2. 给编译器发现的错误做英文注释，如果这文本以／W／开始，那么它只是一个警告；如果不以／W／开始，这是一个足够严重的错误，禁止连接和目的程序的执行。

3.（任选的）所列的错误点上的程序元素被列举，高级特征信息的设计是这样，当信息是显然时，不是所有“信息和错误码”的需要

不考虑是否有列表装置，或什么列表装置是实际的，对错误或警告的总数和特征总是在控制台上显示在编译末尾。这允许你对一个COBOL程序作一个简单的变化，没有列表重新编译它，并且仍然知道是否编译器碰到程序中任何有问题的句子。

不经常发生的二个错误信息，并且被打印在控制台上必须被注意。