

APPLE II COBOL—80

使 用 手 册

COBOL—80

用 户 手 册

目 录

第一章	引言	1
第二章	准备工作	1
第三章	编译	3
第四章	装配	5
第五章	运行	6

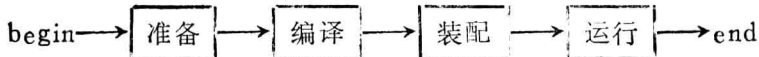
第一章 引言

本手册通过实习指导老师的帮助、指导，通过对外文资料的翻译消化，以及在 Apple—II 机的实践而产生。旨在使用户读完后，即能独立的在 Apple—II 机上研制自己的 MS—COBOL 语言程序。

MS—COBOL 语言是 ANSI、COBOL 的一个子集，用户如果对 MS—COBOL 语言还不熟悉，那就请先阅读一下 MS—COBOL 语言文本。

另外 MS—COBOL 语言是在 CP/M 系统支持下进行工作的，在具体研制 MS—COBOL 语言程序时，还要用到许多 CP/M 的命令。本册在编写时，假定用户对 CP/M 的命令已有所了解。用户如缺乏这方面知识，请参阅。《MS—CP/M USER GUIDE》。

在 Apple—II 机上研制一个 MS—COBOL 程序的大概过程如下：



具体的步骤，在下面几章中详细介绍。

这里假定程序名为 EXAMPLE·COB。当然，用户完全不必拘于此名，而可据自己意愿，任意选择一文件名。

限于水平，错误和不妥之处在所难免，请用户不吝指正。

第二章 准备工作

2.1. 组织磁盘文件

首先，你会得到三个磁盘，其上的文件分配见下图所示。

编译盘（1号盘）

其它文件盘（3号盘）

COBOL COM:COBOL₁·OVR CDANSI·MAC:CDAPPL·REL

COBOL₂ OVR:COBOL₃·OVR COBLIB·REL:CRTEST·COB

COBOL₄ OVR

L80·COM:CDANSI·REL

装配盘（2号盘）

CDAPPL·MAC:SQUARO·COB

COBLBX·REL:COBLOC

CVISAM·COM

M80·COM:LD80·COM

LIB·COM:DEBUG·REL

CREF80·COM:RUNCOB·COM

CRTDRV·REL

在开展工作之前，需要先将原始磁盘做一下拷贝，再把原始磁盘保存起来 以为 备用。然后，再准备一个空盘，作为工作盘，用来存放你的 MS—COBOL 源程序和目标程

序。另外，还要把拷贝得的盘上的文件重新组织一下，以便于以后的工作。具体的做法如下：

一、格式化

- 1.把C P/M系统盘放入驱动器A：，把新盘放入B：。
- 2.打入命令A>FORMAT↵
- 3.根据屏幕的提示信息，通过键盘与之应答，对新盘进行格式化。
- 4.把格式化好的磁盘拿下，换入新盘。重复步骤3。
- 5.重复步骤4，直到格式化完五个新盘。

二、拷贝

- 1.系统盘仍放在驱动器A：中，取一格式化过的盘放入驱动器B：
- 2.打入命令A>COPY┐B:=A:S↵
这样，就拷得了C P/M系统盘，然后对1号、2号、3号盘分别做步骤3、4。
- 3.把原始盘放入驱动器A：，格式化过的空盘放入B：。
- 4.打入命令*┐B:=A:*.*↵

三、组织磁盘文件

现在拷贝工作已经完成。再对磁盘文件重新组织一下，即可开展MS—COBOL程序的研制工作。组织磁盘文件的步骤如下。

首先，产生工作盘。

- 1.把C P/M系统盘放入驱动器A：，把工作盘（此时为格式化过的空盘）放入B：。
- 2.打入命令A>COPY┐B:=A:/S↵
- 3.按回车键退出拷贝程序，并回到操作系统。
- 4.打入命令A>PIP↵
- 5.把系统盘拿下，换上2号盘。
- 6.打入命令*B:RUNCOB.COM=A:RUNCOB.COM↵
这样，工作盘已经产生。屏幕上此时为星号提示符“*”，下面继续组织1号盘、2号盘。

- 1.系统盘仍在驱动器A：中，而把1号盘放入B：
- 2.打入命令*B:ED.COM=A:ED.COM↵
- 3.再把2号盘放到驱动器B：
- 4.按回车键回到操作系统，再打入A>B:↵转连到驱动器B：上。
- 5.打入命令B>ERA┐RUNCOB.COM↵

至此，磁盘准备工作做完，现在磁盘上的文件分配如下。

1号盘	2号盘
COBOL.COM:COBOL ₁ .OVR	COBLBX.REL:LIB.COM
COBOL ₂ .OVR:COBOL ₃ .OVR	COBLOC:DEBUG.REL
COBOL ₄ .OVR:ED.COM	M80.COM:CREF80.COM
	LD80.COM:L80.COM
	CRTDRV.REL
3号盘	工作盘

CDANSI·MAC:CDANSI·REL
CDAPPL·REL:CDAPPL·MAC
CRTEST·COB:CVISAM·COM
L80·COM

RUNCOB·COM

2.2. 输入源程序

现在,可以输入源程序了。

把工作盘放入驱动器A:,把1号盘放入驱动器B:,按一下CTRL—C键,注意,每次在驱动器上换盘后,都要按一下 CTRL—C 键!然后连上驱动器B:,并打入命令B>ED A:EXAMPLE·COB 再打入插入命令*I,此时,可输入你所编好的MS—COBOL语言源程序。输完后,按CTRL—Z键,退出插入。此时可用命令*B#T 看一下所输入的程序。最后,用*E命令把程序存到工作盘上,并退出编辑程序,回到操作系统。

如果在下面几步工作中发现程序有错,需要修改,此时可回到输入源程序这一步,打入命令B>ED A:EXAMPLE·COB 再打入命令*#A,然后对程序进行修改,改好后,还是用*E命令把程序存回工作盘,并退出ED,回到操作系统。

2.3. 打印源程序清单

程序建好后,如果想将其打印出,可以这样来做。

- 1.把系统盘放在驱动器A:里,连上驱动器A:按CTRL—C键。
- 2.打入命令A>DDT
- 3.打入命令—SDD2F
- 4.打入命令DD2F 3E 3I
- 5.按CTRL—C键。

然后,按CTRL—P键,连上打印机。退出打印时,再按一次CTRL—P键断开打印机。

有三种方法能打出程序清单。

- 一、在编辑程序ED·COM控制下,打入命令*B#T
- 二、第一趟编译时,打入命令B>COBOL LST:=A:EXAMPLE·COB
- 三、在操作系统控制下时,打入命令A>TYPE EXAMPLE·COB

第三章 编 译

编译工作需要分两趟来做,第一趟打入命令。

B>COBOL TTY:=A:EXAMPLE·COB

这一趟编译是用来检查源程序是否有错的。编译时的出错情况分两种,一种是低级出错,其错误标志给出在所输出的列表中出错的那行下。在工作时,如果发现此种错误,则请

对照下面给出的出错信息表，将其改正。还有一种错是高级诊断出错，其出错信息给出在列表最后，发生此种错误时，只好根据屏幕上所给出的英文解释，来判断其错误性质，继而考虑对它的修改。

程序改好后，要重新进行第一趟编译，直到屏幕上出现信息 “NO Error or Warnings” 时再进行第二趟编译，第二趟编译才产生目标代码。此时可打入命令 B> COBOL = A : EXAMPLE ↗这样就完成编译工作。

编译出错信息表

出错标志	出错原因	编译程序所采取的补救工作
“QLIT” ?	引用文字错	
	1.零长	忽略且继续编译
	2.连续不恰当	采用适当的
	3.文件结束过早（在结束线前就结束）	采用程序结束
LENGTH?	引用文字长度超过 120 个字符或数字超过18位，或“字”(区别标志或名字)超过30个字符	略去过多的字符
CHRCTR?	非法字符	忽略且继续编译
RUNCT?	标点符号不恰当	采用合适的
BADWORD	当前字畸形，如结束符出现在连字符中或在小数中出现了过多的小数点	忽略且继续编译
NAME?	非法拷贝（COPY）文件名	忽略且继续编译
PIC = X	一个不合适的字型描述	采用PICX
COL. ?	在源程序的第 7 列上出现了不适当的字符，那儿只允许出现字符 * - / D	给第 7 列一个空白
AREA A?	8 ~12列的区域 A 在继续行中不是空白	忽略区域 A 的目标。

编译时还会发生两种错误，一旦出现，必须予以重视。一种是？ Memory Full 这种错误表明程序太大，需要压缩或分作几个模块分别进行编译。另一种出错信息是？ Compiler Error in Phase n at address XXXX 这种错误表明编译开始发生混乱。出错的原因可能是源程序有毛病，如过多地采用二进制数据；或者是编译程序被破坏。在这种情况下，只好用原始磁盘试试。

在编译 MS—COBOL 语言程序时，可以在命令行中添加一些编译开关，使得命令较简洁明了。每添加一个开关时，先打上一斜杠，然后紧跟着打入开关名字符。各开关对编译过程的作用如下。

开关	作	用
／R	把目标文件写到源程序所在的磁盘上，其文件名与源文件相同，而扩展名却表示是目标文件。	
／L	把列表文件写到源程序所在的磁盘上，其文件名与源文件相同，而扩展名却表示是列表文件。	

- ／P 分配堆栈空间以外的 100 个字节供编译程序使用。／P 开关只在编译过程中发生栈溢出时才使用。
- ／D 不产生调试文件，也不输出源文件行数的目标代码信息。
- ／X 准备一个只能连到 COBLIB·REL 中的运行时间系统上的目标文件，而执行程序文件（RUNCOB·COM）却不被编译过的程序装入。用了此开关后，就可以按照 4.6 版本以前所采用的方式来装配程序。
- ／Fn 在列表文件中产生 FIPS 标志信息。编译程序能够检测出 ANSI 1974 COBOL 标准的 Microsoft 扩充部分的语句使用。编译程序还能检测出一些特殊语句所执行的标准级别。在开关后面的数值表明某一特征层要被从特征表中除去。

n 可能的值及其含义在下面给出

值	含 义
0	列出所有 FIPS 标志特征及 Microsoft 的扩充部分。
1	除去低——中特征（LOW—INTERMEDIATE）
2	除去低——中特征和高——中特征（HIGH—INTERMEDIATE）
3	除去所有另外的扩充部分
4	列出非 FIPS 标志特征。

下面给出几个例子，说明编译开关的使用。

， =EXAMPLE/R	相当于	EXAMPLE=EXAMPLE 或=EXAMPLE
， =B:EXAMPLE/L	相当于	， B:EXAMPLE=B:EXAMPLE
， =B:EXAMPLE/R/L	相当于	B:EXAMPLE, B:EXAMPLE=B:EXAMPLE
=EXAMPLE/L/P	相当于	EXAMPLE, EXAMPLE=EXAMPLE/P

第四章 装 配

MS—COBOL 源程序经过编译后，还需再对其进行装配才能运行。

装配时，要把 2 号盘放在驱动器 B：中，而工作盘放在驱动器 A：中，按一下 CTRL—C 键。然后打入命令 B>LDBO A:EXAMPLE, A:EXAMPLE/N/E 即可完成装配工作，并退出装配程序，回到操作系统。

在装配 MS——COBOL 程序时，也可以使用一些开关，它们的名字及其效果是：

开关	效 果
EXAMPLE/N	在装配过程做完时，此开关使得装配程序把可执行程序以 EXAMPLE 为文件名而保存在磁盘上。
／E	此开关使装配程序完成装配处理，并退出回到操作系统。
／G	完成装配处理，并开始程序的执行。
／R	立即把装配程序恢复成初始状态。此开关的使用效果就好象装配程序

刚被从磁盘上装入内存。

下面，是实际在命令行中使用开关的几个例子。

命令行

效

果

EXAMPLE, EXAMPLE/N/E

装配EXAMPLE·REL文件，将EXAMPLE·COM文件存入磁盘，然后回到操作系统。

EXAMPLE/G

装配EXAMPLE·REL文件，并开始运行。但不把绝对代码程序文件EXAMPLE·COM存入磁盘。

EXAMPLE, SUBPRI, B:SUBPR₂, EXAMPLE/N/E

装配EXAMPLE.REL, SUBPRI·REL, 以及SUBPR₂·REL。并将绝对代码程序EXAMPLE·COM存入磁盘，然后退出，回到操作系统。

第五章 运 行

经过了上述五个步骤后，现在工作盘上有了可执行性程序文件EXAMPLE·COM，下面可以运行了。运行时，先连上驱动器A:，然后打入命令A>EXAMPLE)

在运行时，还可能会有一些错误产生，此时屏幕上会给出信息

XX RUN—TIME—ERR

reason

Line number

Program—id

下面给出了出错信息表，你可对照此表，判定程序出错所在，然后回到输入源程序那一步，对源程序进行修改。改好后，再重复编译，装配，运行等过程。

运行出错信息表

原 因

解

释

REDUNDANT OPEN

试图打开已经打开的文件。

DATA UNAVAILABLE

试图引用没有打开或已到了“ATEND”时的文件中的数据。

SUBSCRIPT FAULT

下标为非法值（通常小于1）。

INPUTOUTPUT

没有恢复I/O错。

NON—NUMERIC DATA

每当数字项的内容与PICTURE中所给出的不相符时，就会发生此种错误。

PERFORM OVERLAP

PERFORM中的非法序列，例如当A段被执行时，它前面的另一个PERFORM又准备开始了；或者PERFORM嵌套级超过22级。

CALL PARAMETERS

调用程序和被调用子程序间的参数值不相符。

ILLEGAL READ	试图读一个在 INPUT 或 I—O 方式下没有打开的文件。
ILLEGAL SRITE	试图写一个没有打开的文件。此文件或者是 OUTPUT 方式下的顺序存取文件, 或者是 OUTPUT 或 I/O 方式下的随机存取文件或动态存取文件。
ILLEGAL REWRITE	试图重写一个记录到在 I—O 方式下没有打开的文件上。
REWRITE; NO READ	在上一次操作是一个不正确读的情况下, 试图重写一个记录至顺序存取文件上。
OBJ.CODE ERROR	遇到一个未定义过的目标程序指令。这种情形仅发生在内存中或磁盘文件上的绝对代码程序被损坏的时候。
FEATURE UNIMPL	调用一个不可执行性的目标程序指令。这种情形仅发生在一个损坏了的目标程序中。
GO TO (NOT SET)	企图执行一个未经初始化的可变性段, 此段中仅含有一个非GO语句。
FILE LOCKED	试图在CLOSE WITH LOCK后面打开文件。
READ BEYOND EOF	在遇到文件结束时, 还企图往下读。
DELETE; NO READ	在最后一次操作不是正确的读时, 企图删除顺序存取文件的一个记录。
ILLEGAL DELETE	I—O上没有打开相对文件。
ILLEGAL START	没有为 INPUT 或 I—O打开文件。
NO CRT DRIVER	执行使用光标位置的ACCEPT或DISPLAY语句时, 没有选好CRT驱动程序。
SEG nn LOAD ERR	在企图装入一个程序段时, 出现了不可变性读。两个数字nn是十六进制的, 它是由十进制的段号减去十进制的49而得, 它们与磁盘上的文件的扩展名(.Vnn)相匹配。
EXAMPLE/G	装配EXAMPLE·REL文件, 并开始运行。但不把绝对代码程序文件EXAMPLE.COM存入磁盘。
EXAMPLE, SUBPR ₁ ,	B: SUBPR ₂ , EXAMPLE/N/E
	装配EXAMPLE.REL, SUBPRI.REL, 以及SUBPR ₂ ·REL. 并将绝对代码程序 EXAMPLE.COM 存入磁盘, 然后退出, 回到操作系统。

