

794230

5087

54084

T-2

IBM PC

使用手册

(下)



中国科学院成都计算机应用研究所

系统工程公司

第五章 IBM PC 的编译型 BASIC..... (1)

第一节 BASIC编译程序及使用... (1)

1. 什么是编译程序..... (1)
 - 1.1 解释程序..... (1)
 - 1.2 编译程序..... (1)
2. BASIC 编译程序..... (1)
3. BASIC 编译使用需求..... (2)
4. 用户程序开发..... (3)
5. 首次使用..... (3)
 - 5.1 单驱动器系统..... (3)
 - 5.3 双驱动器系统..... (3)

第二节 对话实例..... (4)

1. 建立和调试DEMO程序 (4)
2. DEMO程序的编译 (2)
 - 2.1 预备步骤..... (4)
 - 2.2 编译程序的使用..... (5)
3. 连接DEMO 程序..... (7)
4. 运行DEMO 程序 (8)
5. 开发程序的更多信息..... (8)

第三节 源程序的建立与调试 (8)

1. 建立源程序..... (8)
2. 编译元命令..... (9)
3. 利用解释程序进行调试..... (10)

第四节 编译..... (10)

1. 准备步骤..... (10)
2. 编译程序的启动..... (11)
 - 2.1 带提示的编译..... (12)
 - 2.2 使用单一命令行..... (13)
3. 编译程序参数..... (14)
 - 3.1 错误俘获参数..... (14)
 - 3.2 事件俘获参数..... (15)

3.3 扫描约定参数..... (18)

3.4 特殊编码参数..... (18)

3.5 运行模块BASRUN.EXE (18)

3.6 编译程序参数小结..... (19)

4. 编译结束..... (19)

5. 示样的编译清单..... (20)

5.1 Offset列和Data列..... (20)

5.3 源程序行..... (20)

5.3 程序清单示例..... (20)

5.4 编译程序信息..... (22)

5.5 小结..... (22)

第五节 程序的连接与运行..... (22)

1. 预备步骤..... (23)

2. 启动连接程序..... (23)

3. 连结工作结束..... (24)

4. 运行一个程序..... (24)

4.1 使用 BASRUN.EXE 的考虑..... (25)

5. 批文件的使用..... (26)

第六节 编译程序与解释程序的差异..... (27)

..... (27)

1. 编译元命令..... (27)

\$INCLUDE..... (28)

\$LINESIZE..... (29)

\$LIST..... (29)

\$OCODE..... (29)

\$PAGE..... (30)

\$PAGEIF..... (30)

\$PAGESIZE..... (30)

\$SKIP..... (30)

\$SUBTITLE..... (31)

\$TITLE..... (31)

2. 操作上的差异..... (31)

3. 语言上的差异..... (32)

CALL语句	(32)
CHAIN语句	(34)
CLEAR语句	(35)
COMMON语句	(35)
DEF FN语句	(36)
DEF type语句	(36)
DIM语句	(37)
DRAW语句	(37)
END语句	(38)
FOR和NEXT语句	(38)
FRE函数	(39)
KEY语句	(39)
OPEN语句	(40)
OPEN # COM语句	(40)
PLAY语句	(42)
REM语句	(43)
RUN语句	(43)
STOP语句	(43)
STRIG函数	(44)
USR函数	(44)
VARPTR函数	(45)
WHILE和WEND语句	(45)
WIDTH语句	(46)
4. 其它差异	(46)
4.1 双精度算术函数	(46)
4.2 双精度循环控制变量	(46)
4.3 表达式计算	(46)
4.4 输入语句	(47)
4.5 整型变量	(47)
4.6 行编辑程序	(48)
4.7 文件的个数	(48)
4.8 PEEK和POKE	(48)
4.9 字符串长度	(48)
4.10 字符串空间分配的实现	(48)
第七节 补充材料	(49)
1. 信息	(49)
1.1 编译错误	(46)
1.2 运行程序错误	(53)
2. 连接程序	(59)
3. 内存分配图	(60)

3.1 段分配图	(60)
3.2 有运行时间模块的内存分配图	(61)
3.3 无运行时间模块的内存分配图	(61)

第六章 FORTRAN (62)

第一节 绪论 (63)

1. FORTRAN 程序结构	(63)
1.1 字符集	(63)
1.2 行	(64)
1.3 列	(64)
1.4 起始行	(64)
1.5 空格	(64)
1.6 注释行	(65)
1.7 标号	(65)
1.8 继续行	(65)
1.9 语句	(65)
1.10 程序单位	(65)
1.11 主程序和子程序	(65)
1.12 语句顺序	(66)
2. 数据类型	(66)
2.1 整型	(67)
2.2 实型	(67)
2.3 逻辑型	(68)
2.4 字符型	(68)
3. 表达式	(68)
3.1 算术表达式	(69)
3.2 字符表达式	(70)
3.3 关系表达式	(71)
3.4 逻辑表达式	(71)
3.5 数组元素名	(72)
3.6 函数引用	(72)
3.7 表达式的优先级	(73)
3.8 表达式的计算规则和限制	(73)
4. FORTRAN名	(73)
4.1 FORTRAN 名作用域	(73)
4.2 未说明的FORTRAN名	(74)

第二节 FORTRAN程序编译 (454)

1. 编译..... (454)	赋值GOTO语句..... (87)
1.1 编译前的准备..... (454)	BACKSPACE语句..... (87)
1.2 启动编译; FOR1..... (454)	块IF语句..... (88)
1.3 继续编译; FOR2..... (454)	CALL语句..... (88)
1.4 连接..... (455)	CLOSE语句..... (89)
1.5 可选择的FOR1命令行... (455)	COMMON语句..... (89)
1.6 可选择的FOR2命令行... (456)	计算GOTO语句..... (90)
1.7 使用批文件进行编译..... (456)	CONTINUE语句..... (91)
1.8 编译大型程序..... (457)	DATA语句..... (91)
2. 设备标识..... (458)	DIMENSION语句..... (92)
3. 编译信息..... (458)	DO语句..... (92)
第三节 编译元命令..... (74)	ELSE语句..... (94)
\$DEBUG..... (75)	ELSEIF语句..... (94)
\$DO66..... (75)	END语句..... (95)
\$INCLUDE..... (76)	ENDFILE语句..... (95)
\$LINESIZE..... (77)	ENDIF语句..... (95)
\$LIST..... (77)	EQUIVALENCE语句..... (96)
\$NODEBUG..... (77)	EXTERNAL语句..... (97)
\$NOLIST..... (77)	FUNCTION语句..... (97)
\$PAGE..... (78)	IMPLICIT语句..... (98)
\$PAGESIZE..... (78)	INTRINSIC语句..... (99)
\$STORAGE..... (78)	逻辑IF语句..... (99)
\$SUBTITLE..... (78)	OPEN语句..... (100)
\$TITLE..... (78)	PAUSE语句..... (102)
第四节 语句..... (79)	PROGRAM语句..... (102)
1. 控制语句..... (79)	READ语句..... (103)
2. 程序、函数和子程序语句... (81)	RETURN语句..... (103)
2.1 主程序..... (82)	REWIND语句..... (104)
2.2 子程序..... (82)	SAVE语句..... (104)
2.3 函数..... (82)	语句函数..... (104)
2.4 形式参数..... (82)	STOP语句..... (105)
3. I/O语句..... (83)	SUBROUTINE语句..... (105)
3.1 I/O语句的元素..... (83)	类型语句..... (106)
3.2 输入输出实体..... (83)	无条件GOTO语句..... (106)
3.3 隐含DO表列..... (84)	WRITE语句..... (107)
4. 说明语句..... (84)	第五节 输入与输出..... (109)
算术IF语句..... (85)	1. 记录..... (109)
赋值语句..... (85)	2. 文件..... (109)
计算赋值语句..... (86)	2.1 文件性质..... (109)
ASSIGN语句..... (86)	2.2 文件名..... (110)

2.3 文件位置..... (110)	2.4 I/O表中的表达式..... (140)
2.4 有格式、无格式和二进制的文件..... (110)	2.5 计算 GOTO 语句中的表达式..... (140)
2.5 顺序存取与直接存取..... (110)	2.6 一般化的I/O..... (141)
2.6 内部文件..... (111)	2.7 对标准FORTRAN的扩充 (141)
2.7 装置..... (111)	2.8 编译程序元命令..... (141)
2.8 概念与限制..... (111)	2.9 反斜杠编辑控制..... (141)
2.9 显式打开外部、顺序有格式文件..... (111)	2.10 文件内部函数的结束... (141)
2.10 其它文件操作..... (112)	3. 连接程序 (LINK)..... (141)
2.11 直接文件与直接设备的联系	3.1 输入文件..... (142)
2.12 BACKSPACE与顺序设备的联系..... (113)	3.2 输出文件..... (142)
2.13 BACKSPACE与无格式顺序文件的联系..... (113)	3.3 VM, TMP (临时文件... (142)
2.14 I/O语句中的函数调用 (113)	3.4 定义..... (142)
2.15 部分读与无格式顺序文件的联系..... (133)	3.5 命令提示..... (143)
3. 有格式I/O与FORMAT语句 (11)	3.6 命令提示的详细描述..... (143)
3.1 格式说明与 FORMAT 语句..... (113)	3.7 怎样启动连接程序..... (146)
3.2 可重复编辑描述符..... (114)	3.8 连接会话的例子..... (148)
3.3 不可重复编辑描述符..... (114)	3.9 怎样确定一个段的绝对地址..... (150)
4. 输入输出表相互作用与格式说明	3.10 信息..... (150)
4.1 输入输出表..... (115)	4. 目标模块的连接..... (152)
4.2 格式说明..... (115)	4.1 与PASCAL的连接..... (153)
5. 编辑描述符..... (115)	4.2 与宏汇编的连接..... (155)
5.1 不可重复描述符..... (115)	5. 实例..... (157)
5.2 可重复描述符..... (117)	
6. 走纸控制..... (118)	
第六节 内部函数..... (118)	
第七节 补充材料..... (120)	
1. 信息..... (120)	
1.1 编译时错误信息..... (120)	
1.2 文件系统错误..... (132)	
2. IBM FORTRAN 与 ANSI FORTRAN 间的差异... (140)	
2.1 完全 FORTRAN 功能... (140)	
2.2 下标表达式..... (140)	
2.3 I/O单元号..... (140)	

第七章 COBOL 语言 (162)

第一节 引言..... (162)

第二节 COBOL程序的编写..... (165)

1. 程序结构..... (165)

1.1 程序的部..... (165)

1.2 编程结构..... (165)

1.3 编程规则..... (166)

1.4 句法表示法..... (167)

1.5 字符集..... (168)

1.6 标点符号..... (169)

1.7 字的格式..... (169)

2. 语句、句子和名字..... (169)

2.1 语句..... (169)

2.2 句子..... (170)

2.3 段.....	(170)	10. 列表和出错信息.....	(191)
2.4 节.....	(170)	11. COPY语句.....	(191)
3. 层号和数据名.....	(170)	12. 实例清单.....	(192)
3.1 记录描述.....	(170)	第四节 标识部	(194)
3.2 数据项.....	(170)	1. AUTHOR段.....	(194)
3.3 数据名.....	(170)	2. DATE _COMPILED段...	(195)
3.4 限定名.....	(173)	3. DATE WRITTEN段 ...	(195)
3.5 条件名.....	(173)	4. IDENTIFICATION DIVISION 标题.....	(195)
3.6 助记名.....	(173)	5. INSTALLATION 段.....	(195)
4. 数据描述项.....	(173)	6. PROGRAM ID 段.....	(195)
4.1 组项.....	(174)	7. SECURITY段.....	(195)
4.2 初等项.....	(174)	第五节 设备部	(196)
5. 文件名.....	(175)	1. 配置节标题.....	(197)
6. 常量.....	(171)	2. 设备部标题.....	(197)
6.1 非数值常量.....	(175)	3. 文件控制段.....	(197)
6.2 数值常量.....	(173)	4. 输入输出节标识.....	(199)
6.3 象征常数.....	(176)	5. I/O控制段.....	(200)
7. 算术表达式.....	(177)	6. 目标计算机段.....	(200)
8. 算术语句.....	(178)	7. 源计算机段.....	(200)
8.1 SIZE ERROR错误.....	(178)	7. 专用名段.....	(200)
8.2 ROUNDED 自选项.....	(179)	第六节 数据部	(202)
8.3 GIVING 自选项.....	(179)	1. 文件节.....	(203)
第三节 COBOL程序的开发	(179)	2. 工作存贮节.....	(203)
1. 程序设计的条件.....	(179)	3. 连接节.....	(204)
2. 编译程序综述.....	(180)	4. 屏幕节.....	(204)
3. 程序开发与源文件.....	(181)	5. 数据部限制.....	(209)
4. COBOL程序的编译.....	(181)	6. 数据部中的子句.....	(209)
4.1 准备工作.....	(181)	BLANK WHEN ZERO子句.....	(209)
4.2 编译步骤.....	(182)	BLOCK 子句.....	(209)
5. COBOL 程序的连接.....	(184)	CODESET子句.....	(209)
6. COBOL 程序的运行.....	(186)	DATA RECORD子句.....	(209)
7. COBOL 自逸命令.....	(186)	文件描述FD项.....	(210)
8. 连接程序自逸命令.....	(187)	JUSTIFIED子句.....	(210)
8.1 例子.....	(188)	LABEL子句.....	(210)
8.2 自动响应文件.....	(188)	LINAGE子句.....	(211)
8.3 子程序的连接.....	(189)	OCCURS子句.....	(211)
8.4 程序段连接.....	(189)	PICTURE子句.....	(213)
8.5 批文件的使用.....	(189)	RECORE子句.....	(216)
8.6 大型程序的编译.....	(190)	REDEFINES子句.....	(217)
9. COBOL使用的文件.....	(190)		

SIGN子句..... (217)	7.1 句法..... (251)
SYNCHRONIZED子句..... (218)	7.2 RECORD KEY子句... (251)
USAGE子句..... (218)	7.3 FILE STATUS报告... (251)
VALUE子句..... (219)	7.4 索引文件的过程部语句... (252)
VALUE OF FILE_ID子句 (220)	8. 语句..... (253)
第七节 过程部 (221)	CLOSE语句..... (253)
1. 说明和使用句子..... (222)	DELETE语句 (索引I/O) (253)
2. 程序分段..... (223)	DELETE语句 (相对I/O) (253)
3. 语句..... (224)	OPEN语句..... (254)
ACCEPT语句..... (224)	READ语句 (索引I/O)..... (254)
ADD语句..... (234)	READ语句 (相对I/O)..... (255)
ALTER语句..... (234)	READ语句 (顺序I/O)..... (256)
COMPUTE语句..... (235)	REWRITE语句 (索引I/O... (257)
DISPLAY语句..... (235)	REWRITE语句 (相对I/O) (257)
DIVIDE语句..... (236)	REWRITE语句 (顺序I/O) (257)
EXHIBIT语句..... (237)	START语句 (索引I/O)..... (257)
EXIT语句..... (237)	START语句 (相关I/O) ... (258)
GOTO语句..... (237)	WRITE语句 (索引I/O) ... (259)
IF 语句..... (238)	WRITE语句 (相对I/O) ... (259)
INSPECT语句..... (240)	WRITE语句 (顺序I/O) ... (259)
MOVE语句..... (241)	第九节 用索引方法进行表处理 (260)
MULTIPLY语句..... (243)	1. 索引名和索引项..... (260)
PERFORM语句..... (243)	SEARCH语句——格式 1 ... (261)
STOP语句..... (244)	SEARCH语句——格式 2 ... (262)
STRING语句..... (244)	SET语句..... (263)
SUBTRACT语句..... (245)	第十节 程序间的通讯 (264)
TRACE语句..... (245)	1. 通讯处理..... (264)
UNSTRING语句..... (246)	1.1 汇编子程序..... (264)
第八节 数据输入与输出 (247)	1.2 例..... (265)
1. 打印文件的处理..... (248)	1.3 连接参数..... (265)
2. 通讯文件的处理..... (248)	CALL语句..... (267)
3. 显示器/键盘输入..... (248)	CHAIN语句..... (267)
4. 软盘文件的处理..... (249)	EXIT PROGRAM语句..... (267)
5. 顺序文件..... (249)	LINKAGE节..... (268)
6. 相对文件..... (249)	带CALL语句和CHAIN 语句的部标题..... (268)
6.1 句法..... (249)	第十一节 补充材料 (268)
6.2 RELATIVE KEY子句 (250)	1. COBOL错误信息..... (268)
6.3 FILE STATUS报告... (250)	1.1 编译时的错误..... (269)
6.4 相对文件的过程部语句 (250)	1.2 运行时错误..... (275)
7. 索引文件..... (250)	

2. COBOL保留字.....	(277)
3. 一个例子.....	(280)
4. 索引文件恢复实用程序.....	(283)

第八章 IBMPC的宏汇编

第一节 引言..... (283)

1. 若干名词解释.....	(284)
2. 宏汇编程序.....	(285)

第二节 相互参照功能..... (289)

1. 相互参照文件的建立.....	(289)
2. 怎样启动CREF程序.....	(289)
2.1 怎样使用自选项1启动 CREF 程 序.....	(290)
2.2 怎样使用自选项2启动 CREF 程 序.....	(290)
2.3 怎样使用自选项3启动CREF 程 序.....	(291)
3. 汇编语言格式.....	(292)
3.1 符号/数据表示.....	(292)
3.2 数据项.....	(292)
3.3 常数.....	(292)
3.4 宏汇编对源程序的扫描.....	(293)
3.5 句子错误.....	(294)
3.6 向前参照值.....	(294)
3.7 标号.....	(294)
3.8 变量.....	(294)
3.9 寄存器代码转换.....	(295)
3.10 标志寄存器.....	(295)
3.11 操作数.....	(296)
3.12 属性操作符.....	(297)
3.13 产生操作符的值.....	(298)
3.14 记录说明操作符.....	(299)
3.15 表达式.....	(300)

第三节 伪指令..... (301)

1. 数据伪指令.....	(301)
ASSUME.....	(301)
COMMENT.....	(302)
DB.....	(302)
DD.....	(303)

DQ.....	(304)
DT.....	(305)
DW.....	(305)
END.....	(306)
EQU.....	(306)
=.....	(307)
EVEN.....	(307)
EXTRN.....	(307)
GROUP.....	(308)
LNCLUDE.....	(310)
LABEL.....	(310)
NAME.....	(311)
ORG.....	(311)
PROC.....	(312)
PUBLIC.....	(313)
.RADIX.....	(313)
RECRRD.....	(313)
SEGMENT.....	(315)
STRUC.....	(317)
2. 条件伪操作.....	(318)
3. 宏伪操作.....	(319)
3.1 宏和重复块伪操作.....	(319)
3.2 宏伪操作详细介绍.....	(323)
MACRO.....	(323)
PURGE.....	(324)
LOCAL.....	(325)
REPT.....	(325)
IRP.....	(326)
IRPC.....	(326)
EXITM.....	(327)
&.....	(327)
;.....	(327)
!.....	(328)
%.....	(328)
4. 列表伪操作.....	(328)
.CREG, .XCREF.....	(329)
.LALL, .SALL, .XALL.....	(329)
LIST, .XLIST.....	(329)
%OUT.....	(329)
PAGE.....	(330)

SUBTTL.....	(330)	IBE, JNA.....	(358)
TITLE.....	(330)	JCXX.....	(359)
4.1 虚假条件块.....	(331)	JE, JZ.....	(359)
4.2 程序员控制.....	(331)	JG, JNLE	(360)
第四节 指令助记符	(333)	JL, JNGE.....	(360)
1. 符号和表示法.....	(333)	JLE, JNG.....	(361)
2. 指令的各个字段.....	(335)	JCXZ.....	(359)
3. 指令详述.....	(336)	JMP.....	(361)
AAA.....	(336)	JNC	(362)
AAD.....	(337)	JNE, JNZ.....	(363)
AAS.....	(337)	JNO.....	(363)
ADC.....	(338)	JNP, JPO.....	(363)
ADD.....	(339)	JNS.....	(364)
AND.....	(340)	JO.....	(364)
CALL.....	(342)	JP, JPE	(364)
CBW.....	(343)	JS	(365)
CLC.....	(344)	LAHF	(365)
CLD.....	(344)	LDS	(365)
CLI.....	(344)	LEA.....	(366)
CMC.....	(344)	LES	(366)
CMP.....	(345)	LOCK	(367)
CMPS.....	(346)	LODS, LODSB, LODSW (367)	
CMPSB.....	(346)	LOOP.....	(369)
CMPSW.....	(346)	LOOPE, LOOPZ.....	(369)
CWD.....	(347)	LOOPNE, LOOPNZ.....	(370)
DAA.....	(348)	MOV.....	(371)
DAS.....	(348)	MOVS, MOVSB和MOVSW (373)	
DEC.....	(349)	MUL.....	(374)
DIV.....	(349)	NEG.....	(375)
ESC (换码)	(351)	NOP.....	(376)
HLT (暂停)	(351)	NOT.....	(376)
IDIV.....	(352)	OR.....	(376)
IMUL.....	(353)	OUT.....	(378)
IN.....	(354)	POP.....	(378)
INC.....	(354)	POPE.....	(379)
INT.....	(355)	PUSH.....	(379)
INTO.....	(356)	PUSHF.....	(380)
IRET.....	(357)	RCL.....	(380)
JA, JARE	(357)	RCR	(381)
JB, JNAE和JC	(388)	REP, REPZ, REPE	(383)

REPNE、REPNZ.....	(383)	3.2 结构类型.....	(411)
RET.....	(384)	3.3 引用类型.....	(413)
ROL.....	(385)	3.4 过程类型.....	(414)
ROR.....	(386)	第六节 变量、表达式与语句 ...	(415)
SAHF.....	(387)	1. 变量.....	(415)
SAL、SHL.....	(387)	1.1 属性.....	(415)
SAR.....	(388)	1.2 属性组合规则.....	(416)
SBB.....	(389)	1.3 VALUE段.....	(416)
SCAS、SCASB、SCASW	(391)	1.4 值.....	(416)
SHR.....	(392)	2. 表达式.....	(417)
STC.....	(393)	2.1 运算符与操作数.....	(418)
STD.....	(393)	2.2 布尔表达式.....	(418)
STI.....	(393)	2.3 集合表达式.....	(418)
STOS、STOSB、STOSW	(394)	2.4 其它表达式.....	(418)
SUB.....	(395)	2.5 函数标志.....	(419)
TEST.....	(396)	3. 语句.....	(419)
WAIT.....	(397)	3.1 语句标号.....	(419)
XCHG.....	(397)	3.2 简单语句.....	(419)
XLAT.....	(398)	3.3 结构语句.....	(421)
XOR.....	(399)	3.4 重复语句.....	(421)
第五节 补充材料	(400)	3.5 顺序控制算符.....	(421)
1. 汇编程序信息.....	(405)	第七节 过程与函数	(422)
2. 输入/输出处理程序信息...	(305)	1. 过程说明与函数说明.....	(422)
3. 运行时的信息.....	(405)	1.1 过程首部与函数首部.....	(422)
第九章 PASCAL语言 (405)		1.2 函数特性.....	(422)
第一节 引 言	(405)	1.3 数据参数.....	(423)
第二节 PASCAL程序的编译 ...	(409)	1.4 内部调用约定.....	(424)
第三节 符号与标号	(407)	2. 可用过程和函数.....	(424)
第四节 编译元命令	(408)	2.1 预先说明的过程和函数...	(424)
第五节 标识符、常数与数据类型 (409)		2.2 数据传送过程和函数.....	(425)
1. 标识符.....	(409)	2.3 算术函数.....	(427)
2. 常数.....	(410)	2.4 REAL函数.....	(428)
2.1 数值常数.....	(410)	2.5 扩展的内容功能特性.....	(428)
2.2 串.....	(410)	2.6 系统内部功能特性.....	(430)
2.3 常数定义.....	(410)	2.7 字符串内部功能特性.....	(431)
2.4 结构常数.....	(410)	2.8 STRING 专用特性.....	(432)
3. 数据类型.....	(411)	2.9 STRING或LSTRING...	(433)
3.1 简单数据类型.....	(411)	2.10 库过程和函数.....	(433)
		第八节 文件系统	(435)
		1. 文件介绍.....	(435)

1.1 文件结构.....	(435)	5. 其它文件过程.....	(447)
1.2 文件方式.....	(436)	6. 文件字段值.....	(448)
2. 文件系统原语.....	(437)	7. 文件F的出错码.....	(449)
2.2 存取缓冲变量.....	(439)	8. 首部中的文件变量.....	(449)
2.2 缓慢求值.....	(439)	10. 系统I/O 特性.....	(450)
3. 文本文件的输入和输出.....	(440)	11. DIRECT文件.....	(450)
4. 扩展I/O功能特性.....	(446)	后记.....	(458)
4.1 暂时文件.....	(447)		

第五章 IBM PC的编译型BASIC

第一节 BASIC编译程序及使用

1. 什么是编译程序

计算机只能执行它自己的机器指令而不可能直接执行 BASIC 语句。因此，在程序运行之前要进行一些翻译工作。把BASIC程序中的语句编译成计算机的机器语言。编译程序和解释程序是实现翻译的二类不同的程序。

1.1 解释程序

用户程序运行时由于解释程序把它逐项翻译成机器语言。为了执行一条 BASIC 语句，解释程序必须分析该语句、检查错误然后执行要求的BASIC功能。

如果一条语句要执行多次（例如，在 FOR...NEXT 循环中），则每执行一次该语句就必须重复一次翻译过程。

此外，BASIC 把用户程序保存为一个编号行的连接表列。这表明计算机不知道各行在内存中的确切位置。当使用 GOTO或GOSUB 等转移到一个具体行时，解释程序必须从头开始查找程序的每一行直至该行号找到为止。

解释程序按类似的方法管理程序中变量的表列。在一个BASIC程序中使用一个变量时，会从头开始查变量表列，直至该变量被找到为止。

上一章叙述的磁带BASIC、磁盘BASIC和高级BASIC都是解释程序。

1.2 编译程序

编译程序与解释程序不同，它把用户写的整个源程序一次翻译并建立一个称为目标文件的文件。目标文件包含的是机器代码。所有翻译工作均在实际运行程序之前完成，在运行程序的过程中不再翻译源程序。此外，内存地址与变量和GOTO 及 GOSUB 对应，因而在程序运行时，不必检查变量表或行号。

有些编译程序称为优化编译程序。它们能够做诸如改变表达式的顺序或消除公共的子表达式等工作以便改善性能或者减少程序的大小。

程序运行时，优化和删除翻译多余步骤的结合会使程序运行得更快。

总的说来，为翻译而输入给编译程序的程序称为源文件，源文件必须是 ASCII 形式。编译程序翻译这个源文件并建立一个新文件作为它的输出。新文件称为浮动目标文件，这二个文件的扩充名分别预置为 .BAS 和 .OBJ。

2. BASIC编译程序

IBM PC BASIC编译程序是一种优化的编译程序，用这种BASIC建立的应用程序

有二个主要的好处。

- 提高大多数程序的执行速度
- BASIC程序的保密

编译后的程序是优化了的机器代码而不是源程序，因而显著地节省执行时间，同时能够保护源程序避免别人未经允许对它进行修改和查看。

BASIC编译程序的另一个优点是，由于BASIC编译程序建立为支持大多数解释型BASIC语言，解释程序与编译程序互为补充。这样，用户就有了一个强有力的程序设计环境，在这种环境中可以用BASIC解释程序迅速运行和调试程序，然后编译这些程序以加快它们的执行速度。

运行时间模块 (Runtime Module)，BASIC编译程序的一个功能是可以建立程序去使用运行时间模块。运行时间模块是名叫BASRUN.EXE的文件，它包含实现BASIC语言需要的大多数例程。可以把它想象成一个子程序库并且它是一个可执行文件。

运行时间模块在程序开始执行时装入，当程序链接至另一个程序时它不重新装入。作为运行时间模块一部分的BASIC例程不需要作为用户编译后产生的可执行程序的一部而保留在软盘上。因此，如果用户建立一个应用程序由几个程序组成，这些程序都使用运行时间模块，则可以减少大量的重复编程并节省软盘空间。

需要BASRUN.EXE运行时间模块的一些应用程序可以向IBM公司订购。当然，不用BASRUN.EXE运行时间模块只用BASIC编译程序照样能开发自己的程序。

BASIC编译程序包的内容，BASIC编译包有下列二张软盘，

- BASIC软盘，它包含文件
 - BASCOM.COM——BASIC编译程序
 - LINK.EXE——连接(LINK)程序
 - DEMO.BAS——演示程序
 - 示范批文件
- LIBRARY软盘，它包含文件
 - BASCOM.LIB——BASIC库
 - BASRUN.EXE——运行时间模块
 - BASRUN.LIB——运行时间模块库
 - IBMCOM.OBJ——通讯模块

3. BASIC编译使用需求

为了在IBM PC上成功地编译BASIC程序，需要。

- BASIC编译程序软盘；BASIC和LIBRARY
- 至少64KRAM
- 一个软盘驱动器（若有二个驱动器操作更方便）
- 一台显示器
- 一台打印机（虽然是任选的，但建议要有打印机。）
- DOS软盘

- 若干空白的51/4"软盘

4. 用户程序开发

用BASIC编译程序开发应用程序的步骤为:

- (1) 建立BASIC源文件
- (2) 调试这个程序
- (3) 编译这个程序
- (4) 连接用户程序所需的模块
- (5) 运行用户程序

这些步骤在后面段落中详细解释,但首先要记住为自己的软盘做备份,然后我们将引导你走一遍使用编译程序的示范。

5. 首次使用

警告:

尽可能为你的BASIC编译主软盘, BASIC和LIBRARY 做备份。建议在开始下面的实例会话之前给这些软盘做备份。

为给软盘做备份必须首先格式化空盘,使用带S自选项的DOS FORMAT 命令。它格式化软盘并把DOS系统文件和 COMMAND.COM 拷贝到用户的软盘上。下一步使用DOS的COPY命令把主文件拷贝到备份盘上。详细做法参见第三章。

把主软盘放到安全的地方,用备份拷贝做工作。

5.1 单驱动器系统

工作时使用下列软盘:

- BASIC软盘
 - LIBRARY软盘
 - 工作软盘——一个格式化的软盘,它将容纳下述文件:
 - 用户源文件
 - 连接步骤中要使用的库 (BASCOM.LIB或BASRUN.LIB)
 - 可能要连到用户程序的其它模块
 - (任选的)运行文件软盘——编译大程序时,可能需要把运行文件写到分开的软盘上。这个软盘将包含:
 - 由连接程序建立的可执行的运行文件
 - BASRUN.EXE,如果编译的程序使用运行时间模块的话。
- 如果不使用运行文件软盘,编译程序和连接程序的所有输出文件通常都到工作软盘。

5.2 二个驱动器系统

有可能发现建立二个新软盘是方便的:一个是用连接程序 (BASIC 软盘的

LINK.EXE)和BASIC库(LIBRARY软盘的BASCOM.LIB)。另一个用连接程序和运行时间模块库(LIBRARY软盘的BASRUN.LIB)

工作时使用下列软盘

- BASIC软盘
- 库连接程序软盘——上述二个软盘中合适的一个,它包含:

LINK.EXE

在连接步骤中要使用的库(BASCOM.LIB或BASRUN.LIB)

- 工作软盘——一个格式化的软盘,它将容纳下述文件;

用户原文件

可能要连到用户程序的其它模块

- (任选的)运行文件软盘——编译大程序时,可能需要把运行文件写到分开的软盘上。

这个软盘将包含:

由连接程序建立的运行(可执行)文件。

BASRUN.EXE,如果属译的程序使用运行时间模块的话。

如果不使用运行文件软盘。编译程序和连接程序的所有输出文件通常都到工作软盘。

5.3 使用前的准备

建议用户在编译任何其它程序之前先编译演示程序,因为这个对话实例将使用户了解编译处理的全貌。此外,还应当读下面各节,它们包含对于成功开发程序十分重要的信息。

第二节 对话实例

本节利用演示程序说明使用BASIC编译程序时一步一步的指令。这些指令包括单驱动器系统和双驱动器系统二方面。

如果严格按本节的叙述打入命令会得到使用BASIC编译的一次成功的对话。如果出现问題,应仔细检查和重做每一步。

请记住上一节介绍的开发应用程序的五个步骤:建立源文件,调试,编译,连接,运行。

1. 建立和调式DEMO程序

因为我们在软盘上准备了一个调试过的演示程序(DEMO.BAS),所以你不必在此执行程序开发过程的前二步。保留在软盘上演示程序是ASCII格式,因为编译程序只能读ASCII格式的文件。

2. DEMO程序的编译

2.1 预备步骤

在实际启用 BASIC 编译程序之前，必须准备工作软盘。为此，要有一个空软盘并按下面的步骤为编译程序使用而准备它。有关命令的详细解释参见第三章。

(1) 启动 DOS。

(2) 若工作软盘未曾格式化，现在就格式化它。使用 DOS FORMAT/S 命令，这样 DOS 系统就拷贝到用户的工作软盘上。

(3) 用 DOS COPY 命令把 DEMO.BAS 程序从 BASIC 软盘拷贝到用户的工作软盘上。

(4) 再从 LIBRARY 软盘把名为 BASRUN.EXE 的文件拷贝到用户工作软盘上。编译出的演示程序将使用运行时模块，因而在运行用户程序时软盘上需要有 BASRUN.EXE。

(5) 如果系统只有一个驱动器，应当把名为 BASRUN.LIB 的文件从 LIBRARY 软盘拷贝到用户工作软盘上。

如果系统有二个驱动器，按第一节第 5 段的介绍建立库连接程序软盘。从格式化后的空软盘开始，分别从 LIBRARY 软盘和 BASIC 软盘把 BASRUN.LIB 和 LINK.EXE 拷贝到该软盘上。我们以后称这个软盘为 BASRUN 连接程序软盘。

在这个演示过程中，所有由编译程序和连接程序建立的文件都将放在这个工作软盘上。

2.2 编译程序的使用

按下列步骤编译程序，

(1) 确保 DOS 已启动，且软盘驱动器 A: 是空的。

如果系统有二个驱动器，把预置驱动器改为 B:，为此，打入：
B:

然后把工作软盘插入驱动器 B:

(2) 把 BASIC 软盘的拷贝插入驱动器 A:，BASIC 软盘包含 BASIC 编译程序。

(3) 打入下列内容作为对 DOS 提示的响应：

A: BASCOM

它将 BASIC 编译程序装入用户的计算机。它将显示一段引头的信息，然后显示一个提示。

(4) 如果系统只有一个驱动器，在显示引头信息之后，从驱动器 A 上取走 BASIC 软盘然后插入工作软盘。

(5) BASIC 编译程序显示的第一个提示为：

Source filename [.BAS] :

由于演示程序的名字是 DEMO.BAS，响应提示应当如下：

Source filename [.BAS] : demo

(记住在打入演示程序的名字后应当按 Enter 键。) 这时不必打入扩充名 .BAS，因为编译程序已经把 .BAS 作为预置的扩充名。

(6) 下一个提示为：

Object filename [DEMO.OBJ] :

这时只须按Enter键。这样产生的目标文件的文件名为DEMO.OBJ。

(7) 最后一个提示将为：

```
Source listing [NUL,LST] ;
```

我们想要一份清单，回答：

```
Source listing [NUL,LST] :demo/e
```

编译程序自动把预置的扩充名加到这个文件名上，这样清单文件名字将是DEMO.LST。回答中的/e不是文件名的一部分而是编译程序的特殊参数。

/E和另外一些参数将在后面的“编译程序参数”中讨论。

上述步骤做完时总的屏幕显示为：

```
IBM Personal Computer BASIC Compiler
```

```
(C) Copyright IBM Corp 1982 Version 1.00
```

```
(C) Copyright Microsoft, Inc. 1982
```

```
Source filename [.BAS] :demo
```

```
Object filename [DEMO.OBJ] ;
```

```
Source listing [NUL,LST] :demo/e
```

一旦用户回答了最后一个提示，编译程序就开始工作。编译程序将产生一个浮动目标程并把它放在一个文件中，该文件的名称就是回答提示“Object filename”时规定的名称。

与此同时，清单文件也将写在用户的软盘上，它的名称为回答提示“Source listing”时规定的名称。

(8) 编译结束时，显示如下信息：

```
nnnn Bytes Available
```

```
nnnn Bytes Free
```

```
φ Warning Error(s)
```

```
φ Severe Error(s)
```

控制返回到DOS（数字nnnn依赖于所用计算机的内存大小）。

如果打入命令DIR，将会看到在工作软盘目录中有三个新文件：DEMO.OBJ和DEMO.LST

(9) 显示进行到此时，可以查看或打印出源清单文件（DEMO.LST）。

打印清单文件的一种方法是从DOS中使用TYPE命令。按Ctrl-Prtsc把屏幕输出送到打印机，然后打入TYPEDEMO.LST。该清单文件将同时在打印机上打印和在屏幕上显示。当文件打印完毕后，再按Ctrl-Prtsc使打印机停止打印屏幕上显示的内容。

第二种打印该文件的方法是使用DOS命令COPY DEMO.LST LPT1。

(10) 看过该清单文件之后，应当删除它以便得到更多的可用空间。为此，从DOS打入命令：

```
ERASE DEMO.LST
```

有关编译和连接文件的详细信息将在第四、第五节给出。现在我们准备作下一步