

บทที่ 7

คู่มือการใช้โปรแกรม เชื่อมโยงบนไมโครคอมพิวเตอร์

โปรแกรม เชื่อมโยงที่สร้างขึ้นบนไมโครคอมพิวเตอร์ สำหรับการนำเสนอ เป็นวิทยานิพนธ์นี้ ได้จัดทำโปรแกรม เชื่อมโยงแยกออกเป็น 4 โปรแกรม ซึ่งแต่ละโปรแกรมจะ ใช้สำหรับภาษาโคบอล ซี เบสิก และแอสเซมบลี ตามลำดับ การแยกโปรแกรมเชื่อมโยง ออกเป็นหลายโปรแกรมนั้น เพราะภาษาต่างๆ เหล่านั้น มีวิธีการเรียกโปรแกรมย่อยภาษา แอสเซมบลี และการส่งผ่านข้อมูลไปโปรแกรมย่อยแตกต่างกัน สำหรับชื่อของโปรแกรม เชื่อมโยงของแต่ละภาษา มีดังนี้

API_ASM.OBJ	เป็นโปรแกรม เชื่อมโยงสำหรับภาษาแอสเซมบลี
API_C.OBJ	เป็นโปรแกรม เชื่อมโยงสำหรับภาษาซี
API_COB.OBJ	เป็นโปรแกรม เชื่อมโยงสำหรับภาษาโคบอล
API_BAS.OBJ	เป็นโปรแกรม เชื่อมโยงสำหรับภาษาเบสิก

ดังนั้น เมื่อผู้ใช้งานพัฒนาโปรแกรมและคอมไพล์โปรแกรมแล้ว ก็จะต้องจัดการ LINK โปรแกรมที่เขียนกับโปรแกรมเชื่อมโยงให้ตรงกันด้วย ก่อนที่จะนำโปรแกรมที่ผ่านการ LINK แล้วมาทำการ RUN โปรแกรมเพื่อติดต่อกับโปรแกรมบนเมนเฟรม รายละเอียดการใช้งาน สำหรับแต่ละภาษามีดังนี้

การเรียกโปรแกรมเชื่อมโยงจากภาษาต่าง ๆ บนไมโครคอมพิวเตอร์มีรูปแบบ
ของการใช้คำสั่งดังนี้

ภาษาซี

```
API_C(PARM1, PARM2, PARM3, PARM4, PARM5, PARM6);
```

ภาษาโคบอล

```
CALL APICOB USING PARM1, PARM2, PARM3,  
PARM4, PARM5, PARM6.
```

ภาษาเบสิก

```
CALL APIBAS(PARM1, PARM2, PARM3, PARM4)
```

ภาษาแอสเซมบลี

```
PUBLIC PARM1:BYTE, PARM2:BYTE, PARM3:BYTE  
PUBLIC PARM4:BYTE, PARM5:WORD, PARM6:WORD
```

```
CALL APIASM
```

- โดยที่ PARM1 = ชื่อของโปรแกรมบน เมนเฟรมมีความยาวขนาด 8 ตัวอักษร
 PARM2 = เป็นข้อมูลที่จะส่งไปยังโปรแกรมบน เมนเฟรมมีขนาดไม่เกิน 1908 ตัวอักษร
 PARM3 = เป็นเนื้อที่สำหรับรับข้อมูลที่ส่งมาจากโปรแกรมบน เมนเฟรม มีขนาดไม่เกิน 1919 ตัวอักษร
 PARM4 = RETURN CODE สำหรับค่านี้ แต่ละค่าจะมีความหมาย ดังนี้

RETURN CODE	ความหมาย
0	ไม่มีข้อผิดพลาด
1	ยังไม่ได้กำหนด Configuration File
2	ขาดการติดต่อสื่อสารกับ เมนเฟรม
3	เกิด Time-out ขณะติดต่อกับ เมนเฟรม

PARM5 = เป็นตัวแปรจำนวนเต็มขนาด 2 ไบท์ ใช้สำหรับเก็บความยาวของ PARM2

PARM6 = เป็นตัวแปรจำนวนเต็มขนาด 2 ไบท์ ใช้สำหรับเก็บความยาวของ PARM3

โปรแกรมภาษาต่างๆ ที่เขียนบนไมโครคอมพิวเตอร์ เมื่อเรียกโปรแกรมเชื่อมโยงเพื่อส่งข้อมูลไปยังโปรแกรมบน เมนเฟรมแล้ว ข้อมูลที่ได้รับกลับมานั้นจะต้องทำการตรวจสอบค่าของ PARM4 ด้วยทุกครั้งก่อนที่จะนำค่าของ PARM3 มาใช้งาน เพื่อไม่ให้โปรแกรมที่เขียนเกิดข้อผิดพลาดในการนำข้อมูลไปใช้งาน

สำหรับตัวอย่างในการกำหนดพารามิเตอร์ต่าง ๆ ในแต่ละภาษามีรายละเอียดในการเตรียมพารามิเตอร์ต่าง ๆ และการเรียกใช้โปรแกรมเชื่อมโยงดังรูปที่ 7.1, 7.2, 7.3 และ 7.4

```
#include <stdio.h>
#include <string.h>
#define d1 "HINQCOB "

main()
{
    char *parm1,*parm2,*parm3,*parm4;
    char p1[9] = d1;
    char p2[11];
    char p3[150];
    char p4[2];
    int  parm5,parm6;
    .
    .
    .

    parm1 = p1;
    parm2 = data;                /* parm2 = data will send to host */
    parm3 = p3;
    parm4 = p4;
    parm5 = 10;
    parm6 = 132;
    api_c(parm1,parm2,parm3,parm4,&parm5,&parm6);
    .
    .
    .
}
```

รูปที่ 7.1 แสดงการกำหนดพารามิเตอร์ในภาษาซี

IDENTIFICATION DIVISION.
PROGRAM-ID. DEM.
ENVIRONMENT DIVISION.

DATA DIVISION.
WORKING-STORAGE SECTION.

```
01 WORK-AREA-PARMS.
   05 PARM1          PIC X(8).
   05 PARM2          PIC X(244).
   05 PARM3.
      10 ACCEPT-MSG  PIC X.
      10 CUST-ACC-NO PIC X(10).
      10 FILLER REDEFINES CUST-ACC-NO.
          15 ERR-CODE PIC X(3).
          15 FILLER   PIC X(7).
      10 CUST-BAL    PIC 9(13)V99.
      10 CUST-NAME   PIC X(16).
      10 FILLER      PIC X(191).
   05 PARM4          PIC 9.
   05 PARM5          PIC S9(4) COMP-0.
   05 PARM6          PIC S9(4) COMP-0.
01 WORK-AREA.
   05 WK-ACC-NO     PIC X(10).
```

PROCEDURE DIVISION.
MAIN-ROUTINE.

```
MOVE 'HINQCOB ' TO PARM1.
MOVE WK-ACC-NO  TO PARM2.
MOVE 0          TO PARM4.
MOVE 10         TO PARM5.
MOVE 50        TO PARM6.

CALL "API_COB"
      USING PARM1, PARM2, PARM3, PARM4, PARM5, PARM6.
```

STOP RUN.

รูปที่ 7.2 แสดงการกำหนดพารามิเตอร์ในภาษาโคบอล

```
PARM1$ = "HINQCOB "  
PARM2$ = "???"      ' parm2 = data will send to host  
PARM3$ = SPACE$(135)  
PARM4% = 0
```

```
CALL API(PARM1$, PARM2$, PARM3$, PARM4%)
```

รูปที่ 7.3 แสดงการกำหนดพารามิเตอร์ในภาษาเบสิก

Apr 25 12:07 1988 test.asm Page 1, line 1

```

-----
; Parameter list areas for send/receive data to/from interface program
-----
parm1    db      8 dup (?)
parm2    db     244 dup (?)
parm3    db     256 dup (?)
parm4    db      ?
parm5    dw     10
parm6    dw     43

;
host_prog db     'HINQASM '
        .
        .
        .
        EXTRN   apiasm:far
        PUBLIC  parm1,parm2,parm3,parm4,parm5,parm6

CODESEG  SEGMENT PARA 'CODE'
BEGIN    PROC    FAR
        ASSUME  CS:CODESEG,DS:DATASG,SS:STACKSG,ES:DATASG
        push   ds
        sub    ax,ax
        push   ax
        mov    ax,datasg
        mov    ds,ax
        mov    es,ax
        .
        .
        .
        copy_str host_prog,parm1,1,8      ; move host program name to parm1
        copy_str namefld,parm2,1,10      ; move data to parm2

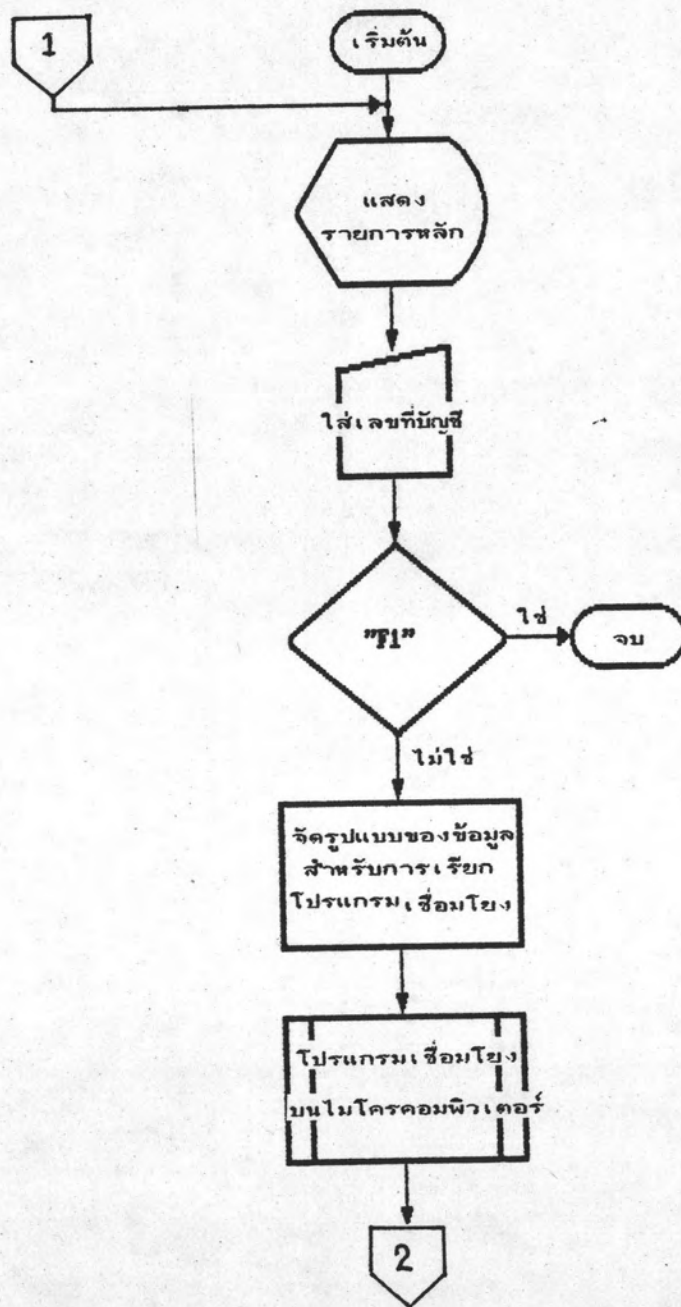
        call   apiasm                    ; call interface program
        .
        .
        .

```

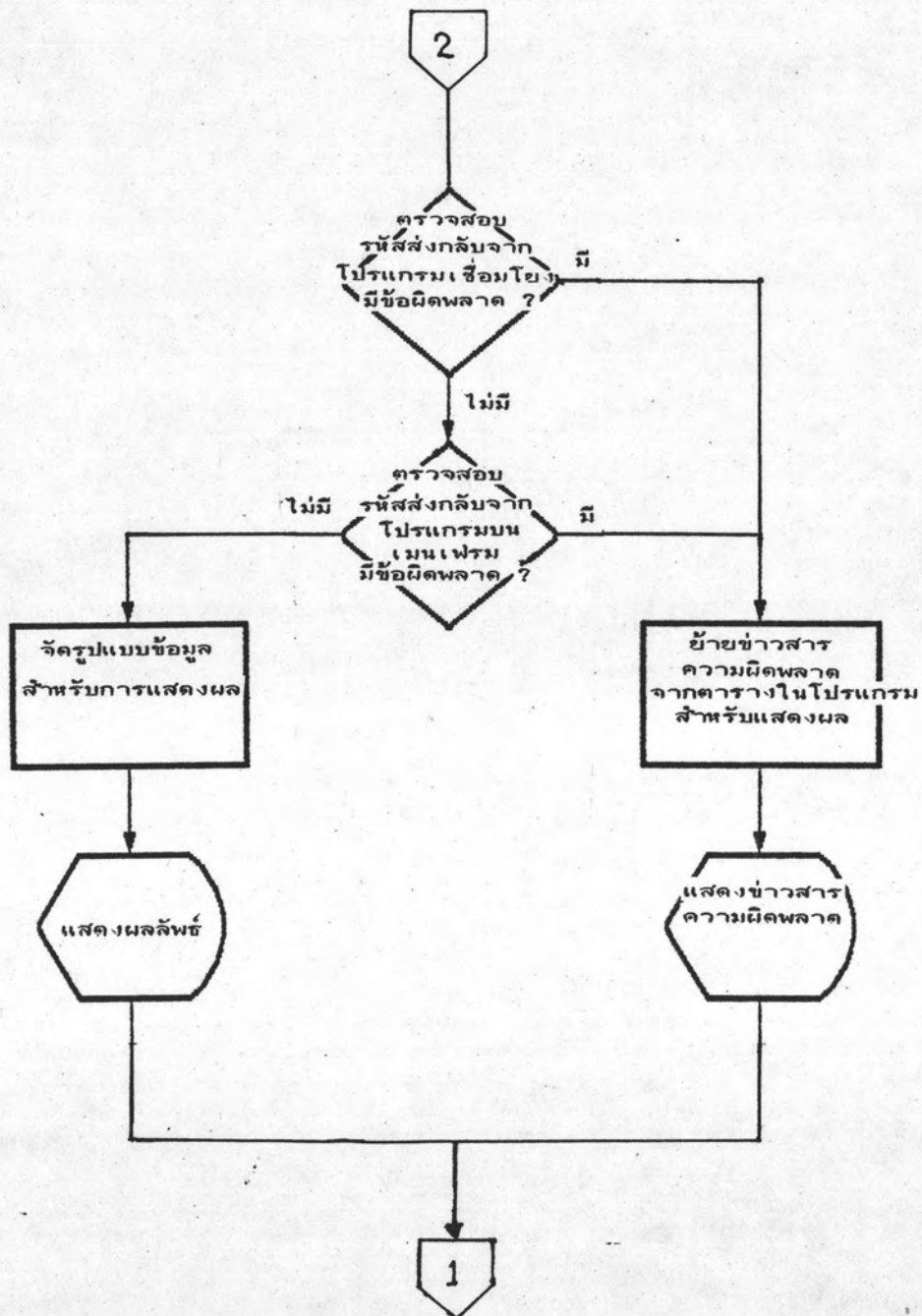
รูปที่ 7.4 แสดงการกำหนดพารามิเตอร์ในภาษาแอสเซมบลี

ตัวอย่างการใช้งานโปรแกรม เชื่อมโยง

สมมติว่าเราจะสร้างโปรแกรมบนไมโครคอมพิวเตอร์ สำหรับสอบถามข้อมูลเกี่ยวกับยอดเงินในบัญชีของลูกค้า ซึ่งข้อมูลเหล่านี้จะเก็บไว้ในแฟ้มข้อมูลบนเมนเฟรม โดยใช้เลขที่บัญชีของลูกค้า (ตัวเลข 10 ตัว) เป็นคีย์ และโปรแกรมบนเมนเฟรมชื่อ "HINQCOB " ทำหน้าที่อ่านข้อมูลจากแฟ้มข้อมูลตามคีย์ที่จะส่งมาจากโปรแกรมบนไมโครคอมพิวเตอร์ และเมื่ออ่านเสร็จก็จะส่งข้อมูลที่ได้อีกกลับมายังโปรแกรมบนไมโครคอมพิวเตอร์ การเขียนโปรแกรมสำหรับภาษาต่าง ๆ แสดงในรูปที่ 7.6, 7.7, 7.8 และ 7.9 และผังการทำงานของโปรแกรมแสดงในรูปที่ 7.5



รูปที่ 7.5 แสดงผังงานของโปรแกรมบนไมโครคอมพิวเตอร์



รูปที่ 7.5 แสดงผังงานของโปรแกรมบนไมโครคอมพิวเตอร์ (ต่อ)

```

#include <stdio.h>
#include <string.h>
#define d1 "HINQCOB "
#define d2 ""
#define d3 ""
#define d4 "E"
#define zero "0"
#define ESC 27
main()
{
    char *parm1,*parm2,*parm3,*parm4;
    char p1[9] = d1;
    char p2[11];
    char p3[150];
    char p4[2] = d4;
    char accno[11],data[11];
    char accbal[16];
    char accname[17];
    char buffer[180];
    long float a;
    int j,i,code;
    int parm5,parm6;

/* ----- */
/* Display menu screen */
/* ----- */
i = 1;
while (i == 1)
{
    CLS();
    setcur( 6,26);printf("Customer Information System");
    setcur( 9,26);printf("          Inquiry          ");
    setcur(12,26);printf("Account No. ");
    setcur(18,24);printf("Please enter your account number");
    setcur(12,38);gets(data);setcur(12,48);

/* ----- */
/* This procedure use for set up parameter and call */
/* interface program. After receive response then */
/* check return code from parameter 4 (parm4) if no */
/* error found (parm4 = 0) then display result */
/* otherwise display error message. */
/* ----- */
    parm1 = p1;
    parm2 = data;
    parm3 = p3;
    parm4 = p4;
    parm5 = 10;
    parm6 = 132;
    api_c(parm1,parm2,parm3,parm4,&parm5,&parm6);
    strncpy(accno,&(p3[0]),10);
    accno[10] = '\0';
    strncpy(accbal,&(p3[10]),15);
    accbal[15] = '\0';
    strncpy(accname,&(p3[25]),16);
    accname[16] = '\0';

```

```
cls();
setcur( 6,26);printf("Customer Information System");
setcur( 9,26);printf("      Inquiry      ");
setcur(12,26);printf("Account No.: ");
setcur(14,26);printf("Name      : ");
setcur(16,26);printf("Balance   : ");
setcur(12,39);printf("%s",accno);
setcur(14,39);printf("%s",accname);
setcur(16,39);printf("%s",accbal);
setcur(24,26);printf("Press ESC for exit to system");
setcur(24,54);code = getch();
if (code == ESC) break;
```

รูปที่ 7.6 แสดงตัวอย่างโปรแกรมภาษาซีในการเรียกโปรแกรมเชื่อมโยง (ต่อ)

IDENTIFICATION DIVISION.
PROGRAM-ID. DEM.
ENVIRONMENT DIVISION.

DATA DIVISION.
WORKING-STORAGE SECTION.

01 WORK-AREA-PARMS.

05 PARM1 PIC X(8).
05 PARM2 PIC X(244).
05 PARM3.
10 ACCEPT-MSG PIC X.
10 CUST-ACC-NO PIC X(10).
10 FILLER REDEFINES CUST-ACC-NO.
15 ERR-CODE PIC X(3).
15 FILLER PIC X(7).
10 CUST-BAL PIC 9(13)V99.
10 CUST-NAME PIC X(16).
10 FILLER PIC X(191).
05 PARM4 PIC 9.
05 PARM5 PIC S9(4) COMP-0.
05 PARM6 PIC S9(4) COMP-0.

01 WORK-AREA.

05 WK-ACC-NO PIC X(10).
05 WK-BAL PIC ZZ,ZZZ,ZZZ,ZZZ,ZZ9.99.
05 TERMINATE-KEY PIC 99.
05 pause pic x.
05 CUST-BAL-ED PIC 9(10)V99.
05 ACC-KEY PIC X VALUE 'A'.

SCREEN SECTION.

01 BLANK-SCREEN.

05 BLANK SCREEN.

01 MENU-SCREEN.

05 BLANK SCREEN.
05 LINE 6 COLUMN 26 HIGHLIGHT VALUE
'Customer Information System'.
05 LINE 9 COLUMN 26 VALUE
' Inquiry '.
05 LINE 12 COLUMN 26 VALUE
'Account No. '.
05 COLUMN PLUS 1 PIC X(10) TO WK-ACC-NO.
05 LINE 18 COLUMN 24 VALUE
'Please enter your account number'.
05 LINE 24 COLUMN 34 HIGHLIGHT VALUE
'F1 ==> Quit'.

01 RESULT-SCREEN.

05 BLANK SCREEN.
05 LINE 6 COLUMN 26 HIGHLIGHT VALUE
'Customer Information System'.
05 LINE 9 COLUMN 26 VALUE
' Inquiry '.
05 LINE 12 COLUMN 26 VALUE
'Account No.: '.
05 COLUMN 39 PIC 9(10) USING CUST-ACC-NO.

```

05 LINE 14 COLUMN 26 VALUE
   'Name      : '.
05 COLUMN 39 PIC X(40) USING CUST-NAME.
05 LINE 16 COLUMN 26 VALUE
   'Balance   : '.
05 COLUMN 39 PIC ZZ,ZZZ,ZZZ,ZZ9.99
   USING CUST-BAL-ED.
05 LINE 22 COLUMN 21 VALUE
   'Please press <enter> for return to Menu'.
01 END-SCREEN.
05 BLANK SCREEN.
05 LINE 12 COLUMN 22 HIGHLIGHT VALUE
   'End of Demonstation System'.

```

PROCEDURE DIVISION.

MAIN-ROUTINE.

```

DISPLAY BLANK-SCREEN.
DISPLAY MENU-SCREEN.
ACCEPT MENU-SCREEN.
ACCEPT TERMINATE-KEY FROM ESCAPE KEY.

```

```

IF TERMINATE-KEY = 2
  DISPLAY BLANK-SCREEN
  DISPLAY END-SCREEN
  STOP RUN.

```

```

MOVE 'HINQCOB ' TO PARM1.
MOVE WK-ACC-NO TO PARM2.
MOVE 0         TO PARM4.
MOVE 10        TO PARM5.
MOVE 50        TO PARM6.

```

```

CALL "API_COB"
      USING PARM1, PARM2, PARM3, PARM4, PARM5, PARM6

```

```

IF PARM4 NOT EQUAL 0
  DISPLAY (24, 30) "Error after call interface program"
  ACCEPT (24, 64) PAUSE
  GO TO MAIN-ROUTINE.

```

```

IF ACCEPT-MSG = ACC-KEY
  MOVE CUST-BAL TO CUST-BAL-ED
  DISPLAY BLANK-SCREEN
  DISPLAY RESULT-SCREEN
  ACCEPT (22, 61) PAUSE
ELSE
  PERFORM ERROR-ROUTINE.
  GO TO MAIN-ROUTINE.

```

ERROR-ROUTINE.

```

IF ERR-CODE = "001"
  DISPLAY (24, 31) "Account not found"
  ACCEPT (24, 49) PAUSE
ELSE
  IF ERR-CODE = "002"

```

```
DISPLAY (24, 30) "Record length error"  
ACCEPT (24, 50) PAUSE  
ELSE  
DISPLAY (24, 30) "Other error on application"  
ACCEPT (24, 54) PAUSE.
```

รูปที่ 7.7 แสดงตัวอย่างโปรแกรมภาษาโคบอลในการเรียกโปรแกรมเชื่อมโยง (ต่อ)

```
-----
MAIN PROGRAM
-----
```

```
-----
This procedure use for initial value
-----
```

```
KEY OFF
KEY 1, "Quit" + CHR$(13)
ON KEY(1) GOSUB Quit
KEY(1) ON
```

```
-----
This procedure use for display Menu screen for enter input data
-----
```

```
Start:
```

```
cls
LOCATE 6,26,1,4,5 : PRINT "Customer Information System"
LOCATE 9,26       : PRINT "      Inquiry      "
LOCATE 12,26      : PRINT "Account No. "
LOCATE 18,24      : PRINT "Please enter your account number."
```

```
-----
This procedur use for get account number from keyboard
-----
```

```
A = 12
B = 38
S$ = ""
Z$ = ""
X$ = ""
A$ = ""
I% = 1
WHILE I% < 11
  J = 1
  LOCATE 12,38,0
  PRINT "      "
  LOCATE 12,38,0
  PRINT A$;
  WHILE J = 1
    LOCATE A,B,1
    S$ = INKEY$
    IF S$ >= "0" AND S$ <= "9" THEN
      A$ = A$ + S$ : PRINT S$
      B = B + 1
      J = 0
      I% = I% + 1
    END IF
    IF (S$ = CHR$(8) OR S$ = CHR$(127)) AND I% > 1 THEN
      B = B - 1
      A$ = LEFT$(A$,LEN(A$)-1)
      I% = I% - 1
      J = 0
    END IF
  WEND
  LOCATE 12,48 : Z$ = INKEY$
```

รูปที่ 7.8 แสดงตัวอย่างโปรแกรมภาษาเบสิกในการเรียกโปรแกรมเชื่อมโยง


```

      WHILE NOT INSTAT : WEND
-----
This procedure use for call interface program to communicate with
application program on Host computer (run under CICS/VS)
-----
      PARM1$ = "HINQCOB "
      PARM2$ = A$
      PARM3$ = SPACES(256)
      PARM4% = 0
      CALL API(PARM1$,PARM2$,PARM3$,PARM4%)
      IF PARM4% <> 0 THEN
          E$ = ""
          LOCATE 22,25,0,4,5 : PRINT "Error from API message no. "
          LOCATE 22,52,0      : PRINT PARM4%
          LOCATE 22,55,1      : E$ = INKEY$
          WHILE NOT INSTAT : WEND
          GOTO start
      END IF
      ACCEPT$ = LEFT$(PARM3$,1)
      IF ACCEPT$ = "E" THEN
          GOSUB ErrorRoutine
          GOTO Start
      END IF
      ACC$ = MID$(PARM3$,2,10)
      B$ = SPACES(12)
      B$ = MID$(PARM3$,15,12)
      BAL = VAL(B$)/100
      NAM$ = MID$(PARM3$,27,16)
      CLS
      LOCATE 6,26,0,4,5 : PRINT "Cuntomer Information System"
      LOCATE 9,26      : PRINT "          Inquiry          "
      LOCATE 12,26     : PRINT "Account No.: "
      LOCATE 12,39     : PRINT ACC$
      LOCATE 14,26     : PRINT "Name          : "
      LOCATE 14,39     : PRINT NAM$
      LOCATE 16,26     : PRINT "Balance       : "
      LOCATE 16,39     : PRINT USING "###,###,###,###.###";BAL
      LOCATE 23,24     : PRINT "Press any key for return to Menu" +CHR$(7)
      LOCATE 23,56,1   : x$ = inkey$
      WHILE NOT INSTAT : WEND
      GOTO Start
END

```

```

-----
Error Routine for display error message
-----

```

```

ErrorRoutine:
      E$ = ""
      E$ = MID$(PARM3$,2,3)
      MSGERR$ = ""

      IF E$ = "001" THEN
          MSGERR$ = "Account not found" + CHR$(7)
      ELSEIF E$ = "002" THEN
          MSGERR$ = "Record length error" + CHR$(7)
      ELSE

```

```

        MSGERR$ = "Other error on application" + CHR$(7)
    END IF

    $$ = ""
    L = LEN(MSGERR$)
    P% = (80 - L) / 2
    C% = P% + L + 1
    LOCATE 23,24,0      : PRINT "
    LOCATE 23,P%        : PRINT MSGERR$
    LOCATE 23,C%,1     : $$ = INKEY$
    WHILE NOT INSTAT : WEND
    RETURN

-----
Quit from demonstation program
-----
Quit:
cls
LOCATE 10,30,0 : PRINT "End of Demonstation"
end

-----
This procedure is INLINE assembly language (Interface program)
-----
SUB  API INLINE
    $INLINE "API.BIN"
END SUB

```

รูปที่ 7.8 แสดงตัวอย่างโปรแกรมภาษาเบสิกในการเรียกโปรแกรมเชื่อมโยง (ต่อ)

```

;-----
; Macro for set cursor position to row and column
;-----
cursor    macro    row,col
           mov     dh,row
           mov     dl,col
           mov     ah,02
           mov     bh,00
           int    10h
           endm

;-----
; Macro for display string
;-----
display   macro    string
           mov     dx,offset string
           mov     ah,09h
           int    21h
           endm

;-----
; Macro use for copy string1 to string2
;-----
copy_str  macro    str_1,str_2,str_pos,str_len
           cld
           mov     bx,str_pos - 1
           lea    si,str_1[bx]
           lea    di,str_2
           mov     cx,str_len
           rep movsb
           endm

;-----
; Macro for get input string from keyboard
;-----
get_str   macro    input_string
           lea    dx,input_string
           mov     ah,0ah
           int    21h
           endm

STACKSG   SEGMENT PARA STACK 'STACK'
           DW    64 DUP(?)
STACKSG   ENDS

;
cr        equ     0dh           ; carriage return
lf        equ     0ah           ; line feed
ESC       equ     1bh           ; ESC key

DATASG    SEGMENT PARA 'DATA'
;-----
; Define Input Area for get data from keyboard
;-----

```

```

NAMEPAR LABEL BYTE
MAXLEN DB 11
ACTLEN DB ?
NAMEFLD DB 11 DUP(' '), '$'
PROMPT DB 'NAME?', '$'
;
; -----
; Data for display Menu screen
; -----
;
Menu db ' Customer Information System'
db 3 dup (cr,lf)
db ' Inquiry '
db 3 dup (cr,lf)
db ' Account No. '
db 6 dup (cr,lf)
db ' Please enter your input data', '$'
;
; -----
; Result screen use for display output data
; -----
;
Result db ' Customer Information System'
db 3 dup (cr,lf)
db ' Inquiry '
db 3 dup (cr,lf)
db ' Account No.: '
accno db 10 dup (?)
db 2 dup (cr,lf)
db ' Name : '
accname db 17 dup (?)
db 2 dup (cr,lf)
db ' Balance : '
accbal db 15 dup (?)
db 6 dup (cr,lf)
db ' Press any key for return to Menu'
db 2 dup (cr,lf)
db ' <ESC> ==> exit '
db '$'
;
end_prog db ' End of Demonstation'
db '$'
;
wait_msg db 'wait ....', '$'
;
err_msg db '..... error return code from host = '
err_no db 3 dup (?)
db ' .....'
db '$'
app_ret db ?
;
; -----
; Parameter list areas for send/receive data to/from interface program
; -----
;
parm1 db 8 dup (?)
parm2 db 244 dup (?)
parm3 db 256 dup (?)
parm4 db ?

```

```

parm5      dw      10
parm6      dw      43

;
;
host_prog  db      'HINQASM '
buffer     db      17 dup (?)
mask       db      ' '
DATASG     ENDS

                EXTRN  apiasm:far
                PUBLIC parm1,parm2,parm3,parm4,parm5,parm6
CODESG
BEGIN        PROC   FAR
                ASSUME CS:CODESG,DS:DATASG,SS:STACKSG,ES:DATASG
                push  ds
                sub   ax,ax
                push  ax
                mov   ax,datasg
                mov   ds,ax
                mov   es,ax

init_rtn:
                call  clear                      ; clear screen
                call  Menu_input                 ; display menu & get input
                copy_str host_prog,parm1,1,8     ; move host program name to parm1
                copy_str namefld,parm2,1,10      ; move data for send to host
                call  apiasm                     ; call interface program
                mov   al,00h                     ;
                cmp   al,parm4                   ; have an error after called ?
                je    no_error                   ; no, go display result.
                call  error_rtn                  ; display error message
                jmp   exit_rtn

no_error:
                copy_str parm3,app_ret,1,1
                mov   al,41h
                cmp   al,app_ret
                je    no_err_2
                call  error_rtn
                jmp   exit_rtn

no_err_2:
                call  no_err_rtn

exit_rtn:
                cmp   al,ESC                      ; enter ESC key ?
                jne   init_rtn
                call  clear                      ; clear screen
                cursor 6,0
                display end_prog
                ret

begin       endp

;
Menu_input proc   near
                cursor 6,0
                display Menu
                cursor 12,38
                get_str namepar
                cursor 24,70

```

```

        display wait_msg
        ret
menu_input endp
;
no_err_rtn proc    near
        call edit_rtn                ; go to edit data for display
        call clear                    ; clear screen for display
        cursor 6,0                    ; set cursor to 6,00
        display result                ; display output on screen
        mov     ah,08h                ;
        int     21h                    ; get one character from keyboard
        ret
no_err_rtn endp

;
;   error routine for display error message after call interface program
;
error_rtn proc    near
        copy_str parm3,err_no,2,3    ; get error message no
        cursor 24,15                 ; set cursor for display
        display err_msg               ; display error message
        mov     ah,08h                ;
        int     21h                    ;
        ret
error_rtn endp

;
;   copy & edit data to screen area for display
;
edit_rtn  proc    near
        copy_str parm3,accno,2,10
        copy_str parm3,accname,27,16
        copy_str mask,buffer,1,17
        copy_str parm3,buffer,14,2
        copy_str parm3,buffer+3,16,3
        copy_str parm3,buffer+7,19,3
        copy_str parm3,buffer+11,22,3
        copy_str parm3,buffer+15,25,2
        copy_str buffer,accbal,1,17
;
;   edit balance field for display by delete leading zero.
;
        mov     cx,13
        lea     si,accbal
mvc_zero:
        cmp     byte ptr[si],30h
        je      mvcbk
        cmp     byte ptr[si],','
        jne     ok_ok
mvcbk:
        mov     byte ptr[si],20h
        inc     si
        loop    mvc_zero
;
;   display result after editing
;

```

```
ok_ok:
edit_rtn    ret
            endp

;
;  clear screen routine
;
clear      proc    near
            mov    ax,0600h
            mov    bh,07
            mov    cx,0000
            mov    dx,184fh
            int    10h
            ret
clear      endp
CODESEG    ENDS
end        begin
```

รูปที่ 7.9 แสดงตัวอย่างโปรแกรมภาษาแอสเซมบลีในการเรียกโปรแกรมเชื่อมโยง (ต่อ)