

บทที่ ๔

การเพิ่มประสิทธิภาพการทำงานของอินเตอร์พรีเตอร์ภาษาเปสิด

โดยปกติแล้วการพัฒนาโปรแกรม จะมีการตัดต่อโปรแกรมในส่วนที่ต้องการ บางส่วนของโปรแกรมหนึ่งสามารถนำไปใช้กับอีกโปรแกรมหนึ่งได้ ซึ่งสามารถทำให้การพัฒนาโปรแกรมสะดวกและมีประสิทธิภาพ แต่การปฏิบัติดังกล่าวไม่สามารถกระทำได้กับหน่วยความจำสำรองชนิด เทปคาสเซท เนื่องจากไม่มีคำสั่งในการปฏิบัติ จากปัญหาดังกล่าว จึงได้พัฒนาคำสั่งที่ใช้ในการเมอร์จ (Merge) โปรแกรมขึ้นมาเพื่อใช้งาน เรียกคำสั่งนี้ว่า MERGE

๔.๑ การเมอร์จโปรแกรม

การเมอร์จ คือ "การนำแฟ้มข้อมูล ๒ แฟ้ม ที่แต่ละแฟ้มได้เรียงตามลำดับแล้วมารวมเป็นแฟ้มข้อมูลเดียวกัน"

โดยปกติแล้ว โปรแกรมภาษาเปสิดจะมีหมายเลขบรรทัด ซึ่งเรียงจากน้อยไปหามาก ฉะนั้น เมื่อนำโปรแกรม ๒ โปรแกรมมาทำการเมอร์จกัน หมายเลขบรรทัด ทั้งหมดจะถูกจัดเรียงใหม่ตามลำดับ แต่ถ้าหมายเลขบรรทัดของโปรแกรมทั้ง ๒ ตรงกัน จะนำหมายเลขบรรทัดและคำสั่งของโปรแกรมที่ ๒ มาแทนที่ หมายเลขบรรทัดและคำสั่งของโปรแกรมที่ ๑ ดูรูปที่ ๔.๑

๔.๒ หลักการทำงานของคำสั่ง MERGE

คำสั่ง MERGE ได้พัฒนาขึ้นเป็นคำสั่งหนึ่งในภาษาเปสิด ซึ่งเป็นคำสั่งแบบไคเรคโมด ก่อนจะใช้คำสั่ง MERGE ควรใช้คำสั่ง CLOAD ก่อน เพื่อโหลดโปรแกรมแรกจากเทปคาสเซท เข้ามาในหน่วยความจำเสียก่อน

PROGRAM 1

๔๕

```
10 ' Merging demonstration program
15 PRINT "Program 1"
30 FOR I=1 TO 10
40   PRINT "PROGRAM 1"
50 NEXT I
```

PROGRAM 2

```
16 PRINT "Program 2"
20 PRINT "Hello !"
40   PRINT "PROGRAM 2"
60 END
```

AFTER USING "MERGE" COMMAND

```
10 ' Merging demonstration program
15 PRINT "Program 1"
16 PRINT "Program 2"
20 PRINT "Hello !"
30 FOR I=1 TO 10
40   PRINT "PROGRAM 2"
50 NEXT I
60 END
```

หลักการทำงานของคำสั่ง MERGE มีดังต่อไปนี้

๔.๒.๑ ทำการโหลดโปรแกรมที่ ๒ จากเทปคาสเซตเข้ามาในหน่วยความจำ และนำมาต่อท้ายโปรแกรมแรก ซึ่งถูกโหลดมาก่อนหน้านี้

๔.๒.๒ เมื่อโหลดโปรแกรมที่ ๒ เข้ามาจนครบแล้ว จะทำการนำหมายเลขบรรทัดของโปรแกรมที่ ๒ ตั้งแต่บรรทัดแรกของโปรแกรมมาเปรียบเทียบกับหมายเลขบรรทัดของโปรแกรมที่ ๑ แล้วทำการเมอร์จบรรทัดของโปรแกรมที่ ๒ เข้ากับโปรแกรมแรกตามหลักการเมอร์จ ดังได้กล่าวไว้ในหัวข้อ ๔.๑

๔.๒.๓ ทำการปรับตัวชี้แอดเดรสบรรทัดต่อไป (Link pointer for next line) ของโปรแกรมที่ทำการเมอร์จเรียบร้อยแล้ว

คำสั่ง MERGE เป็นส่วนหนึ่งของอินเตอร์พรีเตอร์ภาษาเบสิกที่ได้พัฒนาเพิ่มเติม การแทนโทเคน MERGE จะใช้ OCCH ในการแทนคำสั่งในหน่วยความจำ ขั้นตอนการทำงานบางส่วนได้ใช้ส่วนของอินเตอร์พรีเตอร์ที่มีอยู่เดิม เพื่อไม่ให้เป็นการเพิ่มเติมแบบซ้ำซ้อน รายละเอียดขั้นตอนการทำงานแสดงไว้ในผังงานที่ ๔.๑.๑ และโปรแกรมแสดงไว้ในรูปที่ ๔.๓

๔.๓ รูปแบบและการทำงานของคำสั่ง MERGE

การเมอร์จโปรแกรม จำเป็นจะต้องมีโปรแกรมแรกอยู่ในหน่วยความจำเสียก่อน จึงจะต้องใช้คำสั่ง CLOAD เพื่อโหลดโปรแกรมแรกเข้าไปในหน่วยความจำ แล้วจึงใช้คำสั่ง MERGE ซึ่งมีรูปแบบดังรูปที่ ๔.๒ และการทำงานของผังงานที่ ๔.๓

CLOAD"Prog01"

๔๗

```
10 ' Merging demonstration program
15 PRINT "Program 1"
30 FOR I=1 TO 10
40 PRINT "PROGRAM 1"
50 NEXT I
```

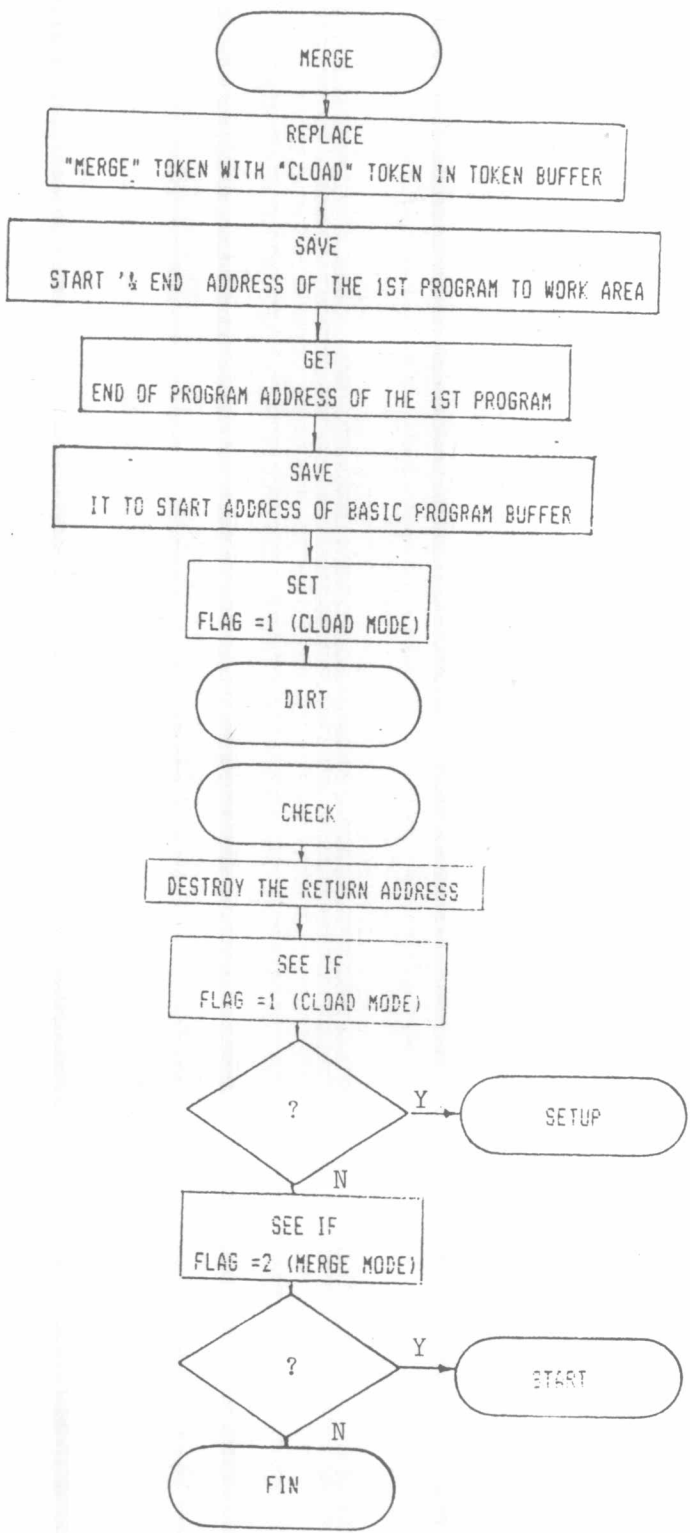
MERGE"Prog02"

```
10 ' Merging demonstration program
15 PRINT "Program 1"
30 FOR I=1 TO 10
40 PRINT "PROGRAM 1"
50 NEXT I
16 PRINT "Program 2"
20 PRINT "Hello !"
40 PRINT "PROGRAM 2"
60 END
```

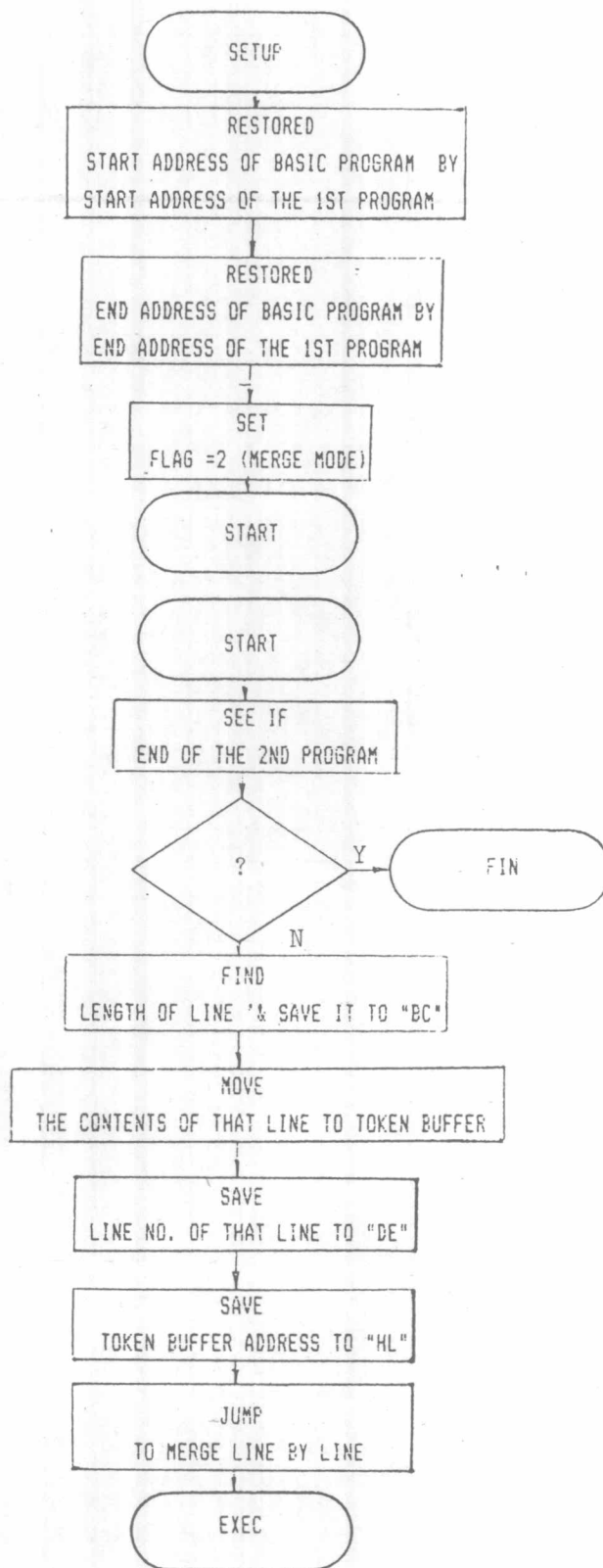
LIST

```
10 ' Merging demonstration program
15 PRINT "Program 1"
16 PRINT "Program 2"
20 PRINT "Hello !"
30 FOR I=1 TO 10
40 PRINT "PROGRAM 2"
50 NEXT I
60 END
```

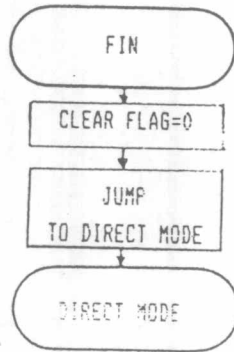
รูปที่ ๔.๒ แสดงรูปแบบของคำสั่ง MERGE



ผังงานที่ ๔.๑ แสดงขั้นตอนการทำงานของคำสั่ง MERGE



ผังงานที่ ๔.๑ แสดงขั้นตอนการทำงานของคำสั่ง MERGE (ต่อ)



ผังงานที่ ๔.๑ แสดงขั้นตอนการทำงานของคำสั่ง MERGE (ต่อ)

UCOM-80 ASSEMBLE LIST ## PAGE 01

```

100      ;           MERGE COMMAND FOR BASIC INTERPRETER
110      ;           BY P.PRAPATSORN
120      ;
130      ;FORMAT :   MERGE "PROG2"
140      ;
150      ;SHOULD BE USE "MERGE" COMMAND
160      ;AFTER LOADED ANY PROGRAMS INTO MEMORY (I.E CLOAD"PROG1")
170      ;
180      ;
190      ORG 0D2E0H
200 B0ED  LDIR: EQU 0B0EDH      ;LDIR
210 52ED  SBCHD: EQU 52EDH      ;SBC HL,DE
220      ;
230 EB54  SBAS: EQU 0EB54H      ;START OF BASIC PROGRAM
240 EFA0  ENDR: EQU 0EFA0H      ;END OF PROG. ADDR.
250 EB57  TKBUF: EQU 0EB57H     ;STARTING TOKEN LINE BUFFER ADDR.
260 3D14  EXEC: EQU 3D14H      ;INSERT LINE ADDR.
270 3D0B  DIRT: EQU 3D0BH      ;DIRECT MODE ADDR.
280 3C9F  OKMOD: EQU 3C9FH     ;OK MODE ADDR.
290      ;
300 D2E0 000000 ADDR: DW 0000H,0000H ;HEADER BUFF:NEXT LINE,LINE#
      D2E3 00
310 D2E4 218000 WORK: DW 8021H,0000H ;WORK AREA,LENGTH OF LINE
      D2E7 00
320 D2E8 9B  CLOAD: DB 9BH      ;CLOAD TOKEN
330 D2E9 00  FLAG: DB 00H      ;FLAG TO FINISH
340      ;
350      ORG 0F13BH
360 F13B C300D3 PT1: JMP MERGE      ;MERGE ROUTINE ADDR.
370      ;
380      ORG 0F17AH
390 F17A C36AD3 PT2: JMP CHECK      ;CLOAD TRAP ROUTINE ADDR.
400      ;
410      ORG 0F16EH
420 F16E C397D3 PT3: JMP STOPK      ;STOP KEY TRAP ROUTINE

```

รูปที่ ๔.๓ โปรแกรมที่ใช้ในการสร้างคำสั่ง MERGE


```

430      ;
440      ;          STARTING PROGRAM
450      ;
460      ORG   0D300H
470 D300 2157EB MERGE: LXI  H,TKBUF
480 D303 3AE8D2      LDA  CLOAD
490 D306 77          MOV  M,A          ;REPLACE 'CLOAD' ON 'MERGE'
500 D307 2A54EB      LHLD SBAS          ;LOAD START ADDR. OF BASIC
510 D30A 22E4D2      SHLD WORK          ;SAVE IT
520 D30D 2AA0EF      LHLD ENDAR          ;GET END OF 1ST PROG ADDR.
530 D310 22E6D2      SHLD WORK+2        ;SAVE IT
540 D313 AF          XRA  A
550 D314 77          MOV  M,A
560 D315 23          INX  H
570 D316 77          MOV  M,A
580 D317 23          INX  H
590 D318 77          MOV  M,A
600 D319 23          INX  H
610 D31A 77          MOV  M,A
620 D31B 2B          DCX  H
630 D31C 2254EB      SHLD SBAS          ;REPLACE ON SBAS
640 D31F 2157EB      LXI  H,TKBUF          ;LOAD TOKEN BUFF. ADDR.
650 D322 2B          DCX  H          ;DECREMENT
660 D323 3E01        MVI  A,01
670 D325 32E9D2      STA  FLAG
680 D328 B7          ORA  A
690 D329 C30B3D      JMP  DIRT          ;JMP TO LOAD TAPE
700      ;
710 D32C 2AE0D2 .START: LHLD ADDR          ;GET STARTING LINE ADDR.
720 D32F 7E          MOV  A,M
730 D330 23          INX  H
740 D331 B6          ORA  M
750 D332 CAA4D3      JZ   FIN          ;IF END OF PROG.2
760 D335 2B          DCX  H
770 D336 11E0D2      LXI  D,ADDR          ;GET STARTING ADDR. OF LINE
780 D339 010400      LXI  B,0004          ;LENGTH OF HEADER OF LINE
790 D33C EDB0        DW   LDIR          ;SAVE HEADER OF LINE
800 D33E E5          PUSH H          ;SAVE ADDR. OF DATA IN LINE
810 D33F 2AE0D2      LHLD ADDR          ;GET STARTING ADDR. OF NEXT LINE
820 D342 D1          POP  D          ;SAVE ADDR. OF DATA IN LINE IN DE
830 D343 ED52        DW   SBCHD          ;FIND LENGTH OF DATA IN LINE
840 D345 E5          PUSH H          ;SAVE LENGTH OF DATA IN LINE
850 D346 23          INX  H          ;FIND LENGTH OF LINE
860 D347 23          INX  H          ;BY ADD UP LENGTH OF HEADER
870 D348 23          INX  H
880 D349 23          INX  H
890 D34A 22E6D2      SHLD WORK+2        ;SAVE LENGTH OF LINE

```

รูปที่ ๔.๓ โปรแกรมที่ใช้ในการสร้างคำสั่ง MERGE (ต่อ)

```

900 D34D C1      POP  B           ;GET LENGTH OF DATA
910 D34E EB      XCHG          ;SOURCE :STARTING LINE ADDR.
920 D34F 1157EB LXI  D,TKBUF   ;DESTINATION :TOKEN LINE BUFFER
930 D352 EDB0    DW    LDIR      ;TRANSFER
940 D354 AF      XRA  A           ;CLEAR ACC.
950 D355 EB      XCHG          ;HL = END OF DATA IN BUFFER
960 D356 77      MOV  M,A         ;PADDED BY ZERO
970 D357 23      INX  H           ;NEXT
980 D358 77      MOV  M,A         ;PADDED BY ZERO
990 D359 2AE2D2  LHLD ADDR+2    ;GET LINE#
1000 D35C EB     XCHG          ;SAVE LINE# IN DE
1010 D35D 2AE6D2 LHLD WORK+2    ;GET LENGTH OF LINE
1020 D360 E5     PUSH H
1030 D361 C1     POP  B           ;SAVE LENGTH OF LINE IN BC
1040 D362 2157EB LXI  H,TKBUF   ;GET TOKEN LINE BUFFER ADDR
1050 D365 2B     DCX  H           ;DECREMENT
1060 D366 B7     ORA  A           ;CLEAR FLAG
1070 D367 C3143D JMP  EXEC      ;JUMP TO INSERT LINE IN PROG.1
1080             ;
1090 D36A E1     CHECK: POP H           ;DESTROY RETURN ADDR.
1100 D36B 21E9D2 LXI  H,FLAG
1110 D36E 35     DCR  M           ;CHECK FLAG
1120 D36F CA79D3 JZ   SETUP    ;IF FLAG=1 -FROM LOAD CASSETE
1130 D372 35     DCR  M           ;CHECK FLAG
1140 D373 CA88D3 JZ   SECND    ;IF FLAG=2 -CONT. TO MERGE
1150 D376 C3A4D3 JMP  FIN      ;ELSE NORMAL MODE
1160             ;
1170 D379 2A54EB SETUP: LHLD SBAS   ;GET STARTING ADDR. OF PROG.2
1180 D37C 22E0D2 SHLD ADDR   ;SAVE TO 'ADDR'
1190 D37F CD90D3 CALL STOP    ;GET STARTING ADDR. OF PROG.1
1200 D382 2AE6D2 LHLD WORK+2  ;GET END ADDR. OF PROG.1
1210 D385 22A0EF SHLD ENDR   ;SAVE IT TO END ADDR.OF BASIC
1220 D388 3E02   SECND: MVI  A,02    ;SET FLAG=2
1230 D38A 32E9D2 STA  FLAG
1240 D38D C32CD3 JMP  START    ;JMP TO MERGE LINE BY LINE
1250             ;
1260 D390 2AE4D2 STOP: LHLD WORK   ;GET STARTING ADDR. OF PROG.1
1270 D393 2254EB SHLD SBAS   ;SAVE TO STARTING BASIC ADDR.

```

รูปที่ ๔.๓ โปรแกรมที่ใช้ในการสร้างคำสั่ง MERGE (ต่อ)

```

1280 D396 C9          RET
1290                ;
1300 D397 CD9ED3 STOPK: CALL RESET          ;RESET FLAG
1310 D39A CD90D3      CALL STOP            ;RESET STARTING ADDR.
1320 D39D C9          RET
1330                ;
1340 D39E 3E00 RESET: MVI A,00              ;SET FLAG=0 -FINISH PROCESS
1350 D3A0 32E9D2      STA FLAG
1360 D3A3 C9          RET
1370                ;
1380 D3A4 CD9ED3 FIN: CALL RESET            ;SET FLAG=0 -FINISH PROCESS
1390 D3A7 C39F3C NORML: JMP OKMOD          ;JMP TO OK MODE
1400                ;
1410 D3AA              END

NO ERRORS
132 STEPS

```

รูปที่ ๔.๓ โปรแกรมที่ใช้ในการสร้างคำสั่ง MERGE (ต่อ)