บทที่ ๒

การทำงานของอินเตอร์เพรตเตอร์

อินเตอร์เพรตเตอร์ภาษาเบสิกเป็นตัวแปลภาษาแบบหนึ่ง ซึ่งทำงานโดยการแปลคำสั่ง (Interpret) ในโปรแกรมเป็นภาษาเครื่องทีละคำสั่ง และปฏิบัติในแต่ละคำสั่งนั้นโดยทันที จนกว่าจะหมดคำสั่งในโปรแกรม แตกต่างจากคอมไพเลอร์ (Compiler) ซึ่งทำงานโดยการแปลคำสั่งทั้งหมดในโปรแกรมเป็นภาษาเครื่องเสียก่อน แล้วจึงปฏิบัติตามคำสั่งจากภาษาเครื่องของโปรแกรมนั้น

๒.๑ ลักษณะการทำงาน (Mode of Operation)

ลักษณะการทำงานของอินเตอร์เพรตเตอร์ แบ่งออกเป็น ๒ แบบ ดังนี้คือ

๒.๑.๑ แบบปฏิบัติทันที (Direct Mode) อินเตอร์เพรตเตอร์จะแปลคำสั่งทีละคำสั่งออกเป็นภาษาเครื่องแล้วปฏิบัติตามคำสั่งนั้นทันที บางคำสั่งเป็นแบบปฏิบัติทันทีเท่านั้น เช่น LIST , NEW , RENUM เป็นต้น

๒.๑.๒ แบบปฏิบัติไม่ทันที (Indirect Mode) คำสั่งที่ต้องการใช้จะอยู่รวมกันเป็นชุด เรียกว่า โปรแกรม (Program) โดยต้องมีเลขหมายบรรทัด (Line number) อยู่ประจำบรรทัดของคำสั่ง คำสั่งในโปรแกรมจะถูกปฏิบัติโดยคำสั่ง 'RUN'

๒.๒ การเปลี่ยนคำสั่งเป็นรหัสในการทำงานของอินเตอร์เพรตเตอร์

คำสั่งในภาษาเบสิกเป็นภาษาชั้นสูงประกอบด้วยตัวอักษรหลายตัว เช่น OPEN , READ , INPUT เป็นต้น การเก็บตัวอักษรลงหน่วยความจำหลักนั้น ทำให้สิ้นเปลืองเนื้อที่ ดังนั้นจึงต้องมีการแปลงคำสั่งออกเป็นรหัสคำสั่ง (Token) เพื่อลดจำนวนตัวอักษร

เนื่องจากคำสั่งในภาษาเบสิกจะขึ้นต้นด้วยตัวอักษร A-Z ดังนั้นสามารถแยกกลุ่มของคำสั่งได้ตามอักษรตัวแรกของคำสั่ง เช่น กลุ่มอักษรตัว A ได้แก่ AND , ABS เป็นต้น อินเตอร์เพรตเตอร์จะเก็บคำสั่งทั้งหลายแยกเป็นกลุ่ม ๆ ตามอักษรตัวแรกนี้ไว้ในตารางคำสั่ง (Statement Table) ข้อมูลอื่นที่จำเป็นในตารางคำสั่งนอกจากคำสั่งก็คือ รหัสคำสั่ง เพื่อให้อินเตอร์เพรตเตอร์แปลงคำสั่งออกเป็นรหัสคำสั่ง

ในการสร้างรหัสคำสั่งนี้ อินเตอร์เพรตเตอร์จะกำหนดรหัสคำสั่งที่มีความหมายโดยใช้ค่าตั้งแต่ 80H-FFH และ FF80H-FFFFH เมื่อนำรหัสคำสั่งนี้ไปผ่านการคำนวณ จะได้ตำแหน่ง (Address Vector) ที่เก็บตำแหน่งการทำงานของคำสั่งนั้น ๆ (Subroutine)

นอกจากนี้ เมื่อพิจารณาดูตารางคำสั่ง จะเห็นว่าคำสั่งแบ่งกลุ่มตามอักษรตัวแรก เพื่อลดขนาดของตารางคำสั่ง อินเตอร์เพรตเตอร์จะตัดตัวอักษรตัวแรกทิ้งในแต่ละคำสั่งที่อยู่ในตารางคำสั่ง ดังนั้นอินเตอร์เพรตเตอร์จำเป็นต้องมีตารางเวกเตอร์ของคำสั่ง (Statement Vector Table) อีกตารางหนึ่งที่เก็บตำแหน่ง (Address) ของคำสั่งแรกของแต่ละกลุ่มคำสั่งในตารางคำสั่ง

๒.๓ ตารางที่จำเป็นสำหรับการเปลี่ยนคำสั่งเป็นรหัส และการทำงาน

๒.๓.๑ ตารางเวกเตอร์ของคำสั่ง (Statement Vector Table)

เป็นตารางที่เก็บตำแหน่งเริ่มต้นของกลุ่มอักษรแต่ละกลุ่มในตารางคำสั่ง ตารางนี้อยู่ที่ตำแหน่ง 021EH - 0251H แต่ละเวกเตอร์ของคำสั่งมีความยาวคงที่ (Fixed Length) คือ ๒ ไบต์ ดังรูปที่ ๒.๑

```
0210                                                    52 02
0220 63 02 6E 02 AD 02 D9 02 FE 02 14 03 28 03 4C 03
0230 6C 03 6D 03 72 03 A6 03 BF 03 DB 03 EE 03 0C 04
0240 0D 04 4B 04 84 04 9B 04 A4 04 BA 04 D2 04 D6 04
0250 D7 04  <------ ตำแหน่งของกลุ่มอักษรตัว 'Z'
```

รูปที่ ๒.๑ แสดงตารางเวกเตอร์ของคำสั่ง

๒.๓.๒ ตารางคำสั่ง (Statement Table)

เป็นตารางที่เก็บคำสั่งทั้งหลายของอินเตอร์เพรตเตอร์ ตารางนี้อยู่ที่ตำแหน่ง 0252H - 04D7H แต่ละคำสั่งในตารางมีความยาวไม่คงที่ (Variable Length) ขึ้นกับจำนวนของตัวอักษรในแต่ละคำสั่ง มีรูปแบบดังนี้ :

[คำสั่งที่ตัดอักษรตัวแรกทิ้ง และบิตแรกของอักษรตัวสุดท้ายเป็น 1 (high bit on) + รหัสคำสั่ง (๑ ไบต์)]

คำสั่งในแต่ละกลุ่มอักษรจะอยู่เรียงลำดับกันไป เมื่อหมดทุกคำสั่งของแต่ละกลุ่มอักษรจะมีค่า 00 ต่อท้ายดังรูปที่ ๒.๒

ในการหาคำสั่งหนึ่ง ๆ อินเตอร์เพรตเตอร์ใช้วิธีเปรียบเทียบทีละคำสั่งเริ่มจากคำสั่งแรกจนกระทั่งคำสั่งสุดท้ายของกลุ่มอักษร สำหรับการเปรียบเทียบแต่ละคำสั่ง อินเตอร์เพรตเตอร์ใช้วิธีเปรียบเทียบทีละไบต์ แต่เนื่องจากรูปแบบของคำสั่งไม่ได้เก็บความยาวของคำสั่งไว้ อินเตอร์เพรตเตอร์จึงใช้บิตแรกของแต่ละไบต์เป็นตัวกำหนด

ความยาวแทน โดยการเปรียบเทียบ ถ้าพบว่าเท่ากันทุกไบต์รวมทั้งไบต์ที่บิดแรกเป็น ๑ ไบต์ถัดไปจะเป็นรหัสคำสั่ง มิฉะนั้นอินเตอร์เพรตเตอร์จะเปรียบเทียบคำสั่งต่อไปจนกระทั่งพบค่า 00 แสดงว่าเกิดความผิดพลาดของคำสั่ง (Syntax Error)

```
0250 D7 04 4E C4 F7 42 D3 06 54 CE 0E 53 C3 14 55 54  ..N..B..T..S..UT
0260 CF A7 00 55 54 54 4F CE 36 45 45 D0 D4 00 4C 4F  ...UTTO.6EE...LO
0270 53 C5 BC 4F 4E D4 98 4C 45 41 D2 92 49 4E D4 1B  S..ON..LEA..IN..
0280 53 4E C7 1C 44 42 CC 1D 56 C9 2A 56 D3 2B 56 C4  SN..DB..V.*V.+V.
0290 2C 4F D3 0C 48 52 A4 15 41 4C CC B1 4F 4D 4D 4F  ,O..HR..AL..OMMO
02A0 CE B3 4B 41 49 CE B4 4F 4C 4F D2 CD 00 41 54 C1  ..HAI..OLO...AT.
02B0 84 49 CD 86 45 46 53 54 D2 A9 45 46 49 4E D4 AA  .I..EFST..EFIN..
02C0 45 46 53 4E C7 AB 45 46 44 42 CC AC 45 C6 96 45  EFSN..EFDB..E..E
02D0 4C 45 54 C5 A6 45 CC A6 00 4E C4 81 4C 53 C5 9E  LET..E...N..LS..
02E0 52 41 53 C5 A2 44 49 D4 A3 52 52 4F D2 A4 52 CC  RAS..DI..RRO..R.
02F0 E5 52 D2 E6 58 D0 0B 4F C6 2E 51 D6 FA 00 4F D2  .R..X..O..Q...O.
0300 82 49 45 4C C4 B9 49 4C 45 D3 BF CE E2 52 C5 0F  .IEL..ILE....R..
0310 49 D8 1E 00 4F 54 CF 89 4F 20 54 CF 89 4F 53 55  I...OT..O T..OSU
0320 C2 8D 45 D4 BA D2 CC 00 4F 4D C5 C7 4C 49 CE CE  ..E.....OM..LI..
0330 47 D2 D1 43 4F 4C 4F D2 D3 50 4C 4F D4 D2 54 41  G..COLO..PLO..TA
0340 C2 C9 53 43 52 CE ED 45 58 A4 19 00 4E 50 55 D4  ..SCR..EX...NPU.
0350 85 C6 8B 4E 53 54 D2 E9 4E D4 05 4D D0 FB 4E 4B  ...NST..N..M..NK
0360 45 59 A4 EE 4E 56 45 52 53 C5 CA 00 00 49 4C CC  EY..NVERS....IL.
0370 C1 00 45 D4 88 49 4E C5 AD 4F 41 C4 BD 53 45 D4  ..E..IN..OA..SE.
0380 C2 50 52 49 4E D4 9B 4C 49 53 D4 9C 50 4F D3 1A  .PRIN..LIS..PO..
0390 49 53 D4 93 4F C7 0A 4F C3 2F 45 CE 11 45 46 54  IS..O..O./E..EFT
03A0 A4 01 4F C6 30 00 45 52 47 C5 BE 4F C4 FC 4B 49  ..O.O.ERG..O..KI
03B0 A4 31 4B 53 A4 32 4B 44 A4 33 49 44 A4 03 00 45  .1KS.2KD.3ID...E
03C0 58 D4 83 4F 52 4D 41 CC CB 4F 54 52 41 43 C5 A0  X..ORMA..OTRAC..
03D0 41 4D C5 C0 45 D7 94 4F D4 E4 00 CE 95 50 45 CE  AM..E..O.....PE.
03E0 B8 D2 F8 43 54 A4 18 50 54 49 4F CE B5 00 55 D4  ...CT..PTIO...U.
03F0 BB 4F 4B C5 97 52 49 4E D4 91 4F D3 10 45 45 CB  .OK..RIN..O..EE.
0400 16 4C 4F D4 D0 44 CC 35 4F D0 AE 00 00 45 41 C4  .LO..D.5O....EA.
0410 87 55 CE 8A 45 53 54 4F 52 C5 8C 45 54 55 52 CE  .U..ESTOR..ETUR.
0420 8E 45 CD 8F 45 53 55 4D C5 A5 53 45 D4 C3 49 47  .E..ESUM..SE..IG
0430 48 54 A4 02 4E C4 08 45 4E 55 CD A8 45 53 45 D4  HT..N..ENU..ESE.
0440 C5 41 4E 44 4F 4D 49 5A C5 B6 00 54 4F D0 90 57  .ANDOMIZ...TO..W
0450 41 D0 A1 41 56 C5 C4 50 43 A8 E3 54 45 D0 E0 47  A..AV..PC..TE..G
0460 CE 04 51 D2 07 49 CE 09 54 52 A4 12 54 52 49 4E  ..Q..I..TR..TRIN
0470 47 A4 E7 50 41 43 45 A4 17 59 53 54 45 CD B7 43  G..PACE..YSTE..C
0480 52 CE EC 00 52 41 43 C5 9F 41 42 A8 DF CF DD 48  R...RAC..AB....H
0490 45 CE DE 41 CE 0D 45 58 D4 C6 00 53 49 4E C7 E8  E..A..EX...SIN..
04A0 53 D2 E1 00 41 CC 13 41 52 50 54 D2 EB 4C 49 CE  S...A..ARPT..LI.
04B0 CF 54 41 C2 C8 50 4F D3 34 00 49 44 54 C8 9D 41  .TA..PO.4.IDT..A
04C0 49 D4 D5 48 49 4C C5 AF 45 4E C4 B0 52 49 54 C5  I..HIL..EN..RIT.
04D0 B2 00 4F D2 F9 00 00 00                          ..O.....
```

รูปที่ ๒.๒ แสดงตารางคำสั่ง

รหัสคำสั่งแบ่งได้ ๒ ประเภทคือ

๒.๓.๒.๑ รหัสคำสั่งประเภทฟังก์ชัน (Function)

หมายถึง รหัสของคำสั่งประเภทฟังก์ชันทั้งหลาย จะทราบได้โดยสังเกตจาก บิตแรกของรหัสจะเป็น 0 (high-bit off) ดังนั้นต้องนำรหัสที่ได้บวกกับค่า FF80H จึงจะเป็นรหัสที่แท้จริงของคำสั่ง

เช่น 06H เป็นรหัสของคำสั่ง ABS มีรหัสที่แท้จริงเป็น FF86H เป็นต้น

๒.๓.๒.๒ รหัสคำสั่งประเภทธรรมดา

หมายถึง รหัสของคำสั่งประเภทธรรมดา จะทราบได้โดยสังเกตจากบิตแรกของรหัสจะเป็น 1 ค่ารหัสนี้เป็นรหัสที่แท้จริงของคำสั่งนั้น

เช่น B1H เป็นรหัสของคำสั่ง CALL เป็นต้น

การคำนวณหาตำแหน่งที่เก็บตำแหน่งการทำงานของคำสั่งมีดังนี้

ตำแหน่ง = 108H + (รหัสคำสั่งในรูปฐานสิบหก - 81H) * 2H

เช่น คำสั่ง CALL มีรหัสเป็น B1H จะมีตำแหน่งของการทำงานของคำสั่งอยู่ที่ 0168H ซึ่งคำนวณจาก 108H + (B1H - 81H) * 2H

๒.๓.๓ ตารางตำแหน่งการทำงานของคำสั่ง
(Subroutine Vector Table)

เป็นตารางที่เก็บค่าตำแหน่งของการทำงานของแต่ละคำสั่ง ตารางนี้อยู่ที่ตำแหน่ง 0108H - 021EH มีความยาวคงที่คือ ๒ ไบต์ สำหรับแต่ละตำแหน่งของคำสั่ง

เช่น จากรหัส B1H ของคำสั่ง CALL นำไปผ่านการคำนวณ จะได้ค่า 0168H ซึ่งที่ตำแหน่ง 0168H ของตารางนี้เก็บค่า 4EAEH ดังนั้นตำแหน่งการทำงานของคำสั่ง CALL เริ่มที่ตำแหน่ง 4EAEH เป็นต้น ดังรูปที่ ๒.๓

```
0100 C3 51 5E F4 2B 55 2C 00 D4 45 AA 12 50 47 CF 15 .Q^.+U,..E..PG..
0110 13 19 AE 3B D2 19 F6 15 60 15 34 15 1F 17 B5 45 ...;....'.4....E
0120 48 15 B4 15 D1 15 CF 45 65 17 C6 46 C6 20 F4 44 H......Ee..F. .D
0130 5D 16 A1 1E CC 22 29 46 92 0D 92 0D 5D 17 C1 20 ]....")F....]..
0140 62 20 D1 15 3D 46 3E 46 43 46 81 46 E8 3E E1 16 b ..=F>FCF.F.>..
0150 A7 16 8A 22 EC 16 FD 22 9F 14 A2 14 A5 14 A8 14 ...".."........
0160 A6 18 96 15 1C 4E 3D 4E AE 4E 06 [illegible] CF 15 37 4F .....N=N.N.R..70
0170 2E 24 78 24 64 5A 71 59 59 55 E0 5B DF 5B 1C 55 .$x$dZqYYU.[.[.U
0180 FE 53 A7 54 B1 5A 1E 59 87 5A BC 55 BB 55 E0 54 .S.T.Z.Y.Z.U.U.T
0190 6E 5A D9 25 C3 25 5B 25 B2 25 CE 25 D1 25 09 26 nZ.%.%[%.%.%.%.&
01A0 5D 26 93 26 AA 26 E1 26 0F 28 0F 28 0F 28 11 27 ]&.&.&.&.(.(.(.'
01B0 3D 27 D3 4A 03 4B 0C 4B FC 2A F8 2C E7 2A 08 39 ='.J.K.K.*.,.*.9
01C0 18 3A B8 3A 4A 29 66 39 B2 3A 45 3B 5A 3B 71 4C .:.:J)f9.:E;Z;qL
01D0 49 1E 67 4A 37 48 2D 48 73 4A 83 4A C2 22 BA 4A I.gJ7H-HsJ.J."J
01E0 2D 48 32 48 44 1E F4 2B 6C 2C 98 2C E5 2C 00 00 -H2HD..+l,.,.,..
01F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0200 00 00 00 00 F8 52 FB 52 FE 52 00 00 C8 56 79 57 .....R.R.R...VyW
0210 90 57 DF 52 E2 52 E5 52 07 27 5E 27 ED 26 52 02 .W.R.R.R.'^'.&R.
0220 63 02 6E 02 AD 02 D9 02 FE 02 14 03 28 03 4C 03 c.n.........(.L.
0230 6C 03 6D 03 72 03 A6 03 BF 03 DB 03 EE 03 0C 04 l.m.r...........
0240 0D 04 4B 04 84 04 9B 04 A4 04 BA 04 D2 04 D6 04 ..K.............
0250 D7 04 4E C4 F7 42 D3 06 54 CE 0E 53 C3 14 55 54 ..N..B..T..S..UT
0260 CF A7 00 55 54 54 4F CE 36 45 45 D0 D4 00 4C 4F ...UTTO.6EE...LO
0270 53 C5 BC 4F 4E D4 98 4C 45 41 D2 92 49 4E D4 1B S..ON..LEA..IN..
0280 53 4E C7 1C 44 42 CC 1D 56 C9 2A 56 D3 2B 56 C4 SN..DB..V.*V.+V.
0290 2C 4F D3 0C 48 52 A4 15 41 4C CC B1 4F [illegible] 4B 4F ,O..HR..AL..OHHO
02A0 CE B3 48 41 49 CE B4 4F 4C 4F D2 CD 00 41 54 C1 ..HAI..OLO...AT.
02B0 84 49 CD 86 45 46 53 54 D2 A9 45 46 49 4E D4 AA .I..EFST..EFIN..
02C0 45 46 53 4E C7 AB 45 46 44 42 CC AC 45 C6 96 45 EFSN..EFDB..E..E
02D0 4C 45 54 C5 A6 45 CC A6 00 4E C4 81 4C 53 C5 9E LET..E...N..LS..
02E0 52 41 53 C5 A2 44 49 D4 A3 52 52 4F D2 A4 52 CC RAS..DI..RRO..R.
02F0 E5 52 D2 E6 58 D0 0B 4F C6 2E 51 D6 FA 00 4F D2 .R..X..O..Q...O.
```

ตำแหน่งการทำงานของคำสั่ง CALL

คำสั่ง CALL มีรหัสคำสั่งเป็น B1H

รูปที่ ๒.๓ แสดงการหาตำแหน่งการทำงานของคำสั่ง 'CALL'

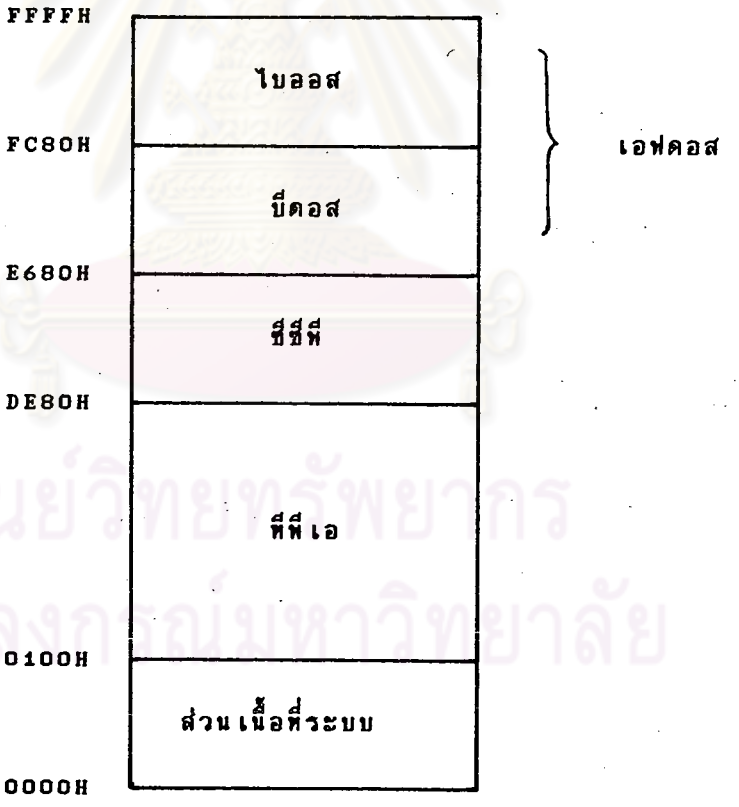
๒.๔ การแบ่งหน่วยความจำหลักของระบบดำเนินการซีพีเอ็ม

(Memory Map of CP/M System)

สำหรับหน่วยความจำหลักมี 64 กิโลไบต์ จากตำแหน่ง 0000H - FFFFH ที่ตำแหน่ง 0000H - 0100H คือส่วนเนื้อที่ระบบ (System Area) เป็นส่วนที่ระบบจะใช้ทำงาน ใต้ FFFFH ลงมา คือส่วนของไบออส (Basic Input/Output System) เป็น

ส่วนที่ทำงานเกี่ยวกับอินพุท (Input) และ/หรือเอาท์พุท (Output) ทางกายภาพ (Physical) ได้ไบออสคือส่วนของบีดอส (Basic Disk Operating System) เป็นส่วนที่ทำงานเกี่ยวกับอินพุทและ/หรือเอาท์พุททางตรรก (Logical) รวมทั้งการจัดการแฟ้มข้อมูลจานแม่เหล็กแบบอ่อน ได้บีดอสคือส่วนของซีซีพี (Console Command Processor) เป็นส่วนที่ใช้ติดต่อกับผู้ใช้ผ่านทางคีย์บอร์ด (Keyboard) ระหว่างได้ซีซีพีและเหนือส่วนเนื้อที่ระบบคือส่วนของทีพีเอ (Transient Program Area) เป็นส่วนที่ใช้เป็นเนื้อที่ของหน่วยความจำหลักสำหรับโปรแกรม ส่วนบีดอสและไบออสรวมเรียกว่าเอฟดอส (FDOS)

การแบ่งหน่วยความจำหลักของระบบดำเนินการซีพีเอ็ม แสดงดังรูปที่ ๒.๔



รูปที่ ๒.๔ แสดงการแบ่งหน่วยความจำหลักของระบบดำเนินการซีพีเอ็ม

๒.๔ การทำงานครั้งแรกของอินเตอร์เพรตเตอร์

อินเตอร์เพรตเตอร์จะถูกโหลด (Load) ลงส่วนที่พีเอ คือถัดจากส่วนเนื้อที่ระบบขึ้นไป หลังจากนั้นอินเตอร์เพรตเตอร์จะทำงาน โดยมีขั้นตอนของการทำงานหลักดังนี้

๒.๔.๑ กำหนดสแต็คของอินเตอร์เพรตเตอร์

๒.๔.๒ คำนวณเนื้อที่ของแฟ้มข้อมูลใช้งาน (File Buffer) ซึ่งจะใช้ที่ต่อจากตัวอินเตอร์เพรตเตอร์

๒.๔.๓ ได้ตำแหน่งที่จะโหลดโปรแกรมลงได้ (Beginning of Program Text Area Address) คือต่อจากเนื้อที่แฟ้มข้อมูลใช้งาน ค่านี้เก็บอยู่ที่ตำแหน่ง 0869H

๒.๔.๔ ส่งข้อความครั้งแรกของอินเตอร์เพรตเตอร์ และค่าจำนวนไบต์ของเนื้อที่ที่ใช้งานได้ทางจอภาพ (Screen)

๒.๔.๕ ส่งข้อความ OK ทางจอภาพเพื่อเตรียมที่จะทำงานต่อไป

ผังงานการทำงานของอินเตอร์เพรตเตอร์ แสดงในภาคผนวก ก.

ถ้ามีโปรแกรมทำงานที่จะต้องโหลด อินเตอร์เพรตเตอร์จะโหลดโปรแกรมนั้นลงมาทำงาน ณ ตำแหน่งที่เก็บไว้ใน 0869H แล้วเริ่มแปลทีละคำสั่ง ถ้าโปรแกรมมีการใช้ตัวแปร ชื่อตัวแปรจะเก็บไว้ในตารางตัวแปร (Variable Table) ซึ่งอยู่ถัดจากโปรแกรมขึ้นไป ตำแหน่งเริ่มต้นของตารางตัวแปร (Beginning of Variable Table Address) เก็บอยู่ที่ตำแหน่ง 0B92H และตำแหน่งว่างที่จะใช้ต่อไปของตารางตัวแปร (Available of Variable Table Address) เก็บอยู่ที่ตำแหน่ง 0B94H

สำหรับค่า (Content) ของตัวแปรสตริง (String) นั้น ในตารางตัวแปรจะเก็บเฉพาะตัวแปรและตำแหน่งที่เก็บค่าตัวแปรนั้น โดยที่ค่าตัวแปรนี้อาจอยู่ในโปรแกรมเอง หรืออยู่ในตารางค่าสตริง (String Table) ซึ่งเก็บอยู่ใต้สแต็ค (Stack) การทำงานของอินเตอร์เพรตเตอร์ซึ่งอยู่ใต้เอฟดอสลงมา ทั้งนี้ตำแหน่งเริ่มต้นของตารางค่าสตริง (Beginning of String Table Address) เก็บอยู่ที่ตำแหน่ง 0B46H และตำแหน่ง

ว่างที่จะใช้ต่อไปของตารางค่าสตริง (Available of String Table Address) เก็บอยู่ที่ตำแหน่ง 0B6BH ดังรูปที่ ๒.๕

รูปที่ ๒.๕ แสดงการแบ่งหน่วยความจำหลักของอินเตอร์เพรตเตอร์

๒.๖ <u>การเก็บตัวแปรในตารางตัวแปร และค่าของตัวแปรในตารางค่าสตริง</u>

(Variable Table & String Table)

ตารางตัวแปร เป็นตารางที่ใช้เก็บตัวแปรและค่าของตัวแปร สำหรับตารางค่าสตริง เป็นตารางที่ใช้เก็บค่าของตัวแปรสตริง

รูปแบบที่เก็บในตารางตัวแปรสำหรับตัวแปรหนึ่ง ๆ คือ

๒.๖.๑ ประเภทของตัวแปร (Type)

ใช้ ๑ ไบต์ มีค่าและความหมายดังนี้

๒ หมายถึง ตัวแปรจำนวนเต็ม (Integer)

๓ หมายถึง ตัวแปรสตริง (String)

๔ หมายถึง ตัวแปรทศนิยม ๗ ตำแหน่ง (Single Precision)

๕ หมายถึง ตัวแปรทศนิยม ๑๖ ตำแหน่ง (Double Precision)

๒.๖.๒ ชื่อของตัวแปร (Name)

มี ๓ กรณีคือ

๒.๖.๒.๑ ถ้ายาวไม่เกิน ๒ ไบต์ จะใช้ ๓ ไบต์ โดยไบต์ที่ ๑ , ๒ เป็นรหัสแอสคีของชื่อตัวแปรนั้น และ ไบต์ที่ ๓ เป็น ๐

เช่น ตัวแปร A จะเก็บเป็น 410000

๒.๖.๒.๒ ถ้ายาวมากกว่า ๒ ไบต์ จะใช้จำนวนไบต์เท่ากับจำนวนของตัวแปร + ๑ โดยไบต์ที่ ๑ , ๒ เป็นรหัสแอสคีของชื่อตัวแปร ๒ ตัวแรก ไบต์ที่ ๓ บอกว่าความยาวที่ยาวกว่า ๒ ไบต์เท่าใด และตั้งแต่ไบต์ที่ ๔ เป็นต้นไปเป็นค่ารหัสแอสคีของตัวแปรที่เหลือ โดยมีบิตแรกของแต่ละตัวอักษรเป็น 1

เช่น ตัวแปร ABCD จะถูกเก็บเป็น 414202C3C4

๒.๖.๓ ค่าของตัวแปร หรือ ตำแหน่งของตัวแปร (Content or Adress)

๒.๖.๓.๑ กรณีที่ไม่ใช่ตัวแปรสตริง

ค่านี้จะเป็นค่าของตัวแปรซึ่งความยาวที่ใช้เก็บเป็นดังนี้

๒ ไบต์ สำหรับตัวแปรจำนวนเต็ม

๔ ไบต์ สำหรับตัวแปรทศนิยม ๗ ตำแหน่ง

๘ ไบต์ สำหรับตัวแปรทศนิยม ๑๖ ตำแหน่ง

๒.๖.๓.๒ กรณีที่เป็นตัวแปรสตริง

ค่านี้จะเป็นตำแหน่งของค่าของตัวแปร ซึ่งความยาวที่ใช้เก็บเท่ากับ ๒ ไบต์ สำหรับค่าของตัวแปรสตริงอาจจะอยู่ในตัวโปรแกรมเองหรืออยู่ในตารางค่าสตริง ขึ้นอยู่กับการเขียนโปรแกรมแต่ละโปรแกรม แต่โดยทั่วไปค่าของตัวแปรสตริงจะอยู่ในตารางค่าสตริง ซึ่งทุกครั้งที่มีการเปลี่ยนแปลงค่าของตัวแปร อินเตอร์เพรตเตอร์จะนำค่าใหม่ไปไว้ยังตำแหน่งใหม่ในตารางค่าสตริง และปรับตำแหน่งของค่าของตัวแปรใหม่ในตารางสตริง

ในกรณีนี้จะเห็นว่า ค่าเดิมของตัวแปรสตริงจะไม่ถูกนำมาใช้งานอีกเลย เรียกค่าเดิมของตัวแปรสตริงว่าเป็นขยะ (Garbage) ของตารางค่าสตริง ซึ่งเมื่ออินเตอร์เพรตเตอร์ทำงานไประยะหนึ่ง จะมีการปรับตารางค่าสตริงให้คงเหลือแต่ค่าที่ใช้งานเท่านั้น โดยปรับค่าของตัวแปรสตริงในตารางค่าสตริง และปรับตำแหน่งค่าของตัวแปรสตริงในตารางตัวแปร การทำงานเช่นนี้จะเรียกว่า การล้างขยะ (Clear Garbage)

๒.๖.๔ ความยาว (Length)

สำหรับกรณีที่เป็นตัวแปรสตริงเท่านั้น เพื่อบอกให้ทราบว่าค่าของตัวแปร มีความยาวเท่าใด

๒.๗ สแต็คของอินเตอร์เพรตเตอร์

สแต็คของอินเตอร์เพรตเตอร์ คือส่วนที่อินเตอร์เพรตเตอร์จะใช้เก็บตำแหน่งต่าง ๆ ของส่วนของการทำงานในระหว่างการทำงาน ใช้ที่ตั้งแต่ใต้เอฟดอสลงมา ผู้ใช้สามารถกำหนดขนาดของสแต็คได้ แต่ถ้าไม่กำหนดอินเตอร์เพรตเตอร์จะใช้ขนาด ๒ เพจ (Pages) โดยที่ ๑ เพจ = ๒๕๖ ไบต์ ตำแหน่งเริ่มต้นของสแต็ค (Bottom of stack) กำหนดโดยอินเตอร์เพรตเตอร์ และตำแหน่งสูงสุดที่จะใช้งานได้ของสแต็ค (Top of stack) โดยคำนวณจากค่าตำแหน่งเริ่มต้นของสแต็ค + ขนาดของสแต็ค

๒.๘ การใช้แฟ้มข้อมูลในโปรแกรม

ถ้าโปรแกรมมีการใช้แฟ้มข้อมูล อินเตอร์เพรตเตอร์จะใช้เนื้อที่ที่ต้องทำงานเกี่ยวกับแฟ้มข้อมูลเรียกว่า เนื้อที่แฟ้มข้อมูลทำงาน ซึ่งจะกินที่ต่อจากตัวอินเตอร์เพรตเตอร์เอง และโปรแกรมจะกินที่ต่อจากเนื้อที่แฟ้มข้อมูลทำงานนี้อีกที

ขนาดของเนื้อที่แฟ้มข้อมูลทำงานขึ้นกับ จำนวนแฟ้มข้อมูลและความยาวสูงสุดของระเบียนของแฟ้มข้อมูลที่สามารถใช้ได้ในโปรแกรมการทำงานขณะนั้น ผู้ใช้สามารถกำหนดขนาดของเนื้อที่แฟ้มข้อมูลทำงานได้ โดยกำหนดจำนวนแฟ้มข้อมูลและความยาวสูงสุดของระเบียนของแฟ้มข้อมูล แต่ถ้าไม่กำหนดอินเตอร์เพรตเตอร์จะใช้เนื้อที่สำหรับ ๓ แฟ้มข้อมูล และความยาวสูงสุดของระเบียน ๑๒๘ ไบต์

แต่ละแฟ้มข้อมูลใช้เนื้อที่ทำงาน = ความยาวสูงสุดของระเบียน + 00B2H

เช่น ค่าความยาวสูงสุดของระเบียน = ๑๒๘ ไบต์
แฟ้มข้อมูลที่ ๑ จะกินที่ตั้งแต่ตำแหน่ง 5EF8H ค่านี้เก็บอยู่ที่ตำแหน่ง 0875H
แฟ้มข้อมูลที่ ๒ จะกินที่ตั้งแต่ตำแหน่ง 602AH ค่านี้เก็บอยู่ที่ตำแหน่ง 0877H
แฟ้มข้อมูลที่ ๓ จะกินที่ตั้งแต่ตำแหน่ง 615CH ค่านี้เก็บอยู่ที่ตำแหน่ง 0879H

๒.๙ การคำนวณหาตำแหน่งเริ่มต้นของโปรแกรม

ตำแหน่งเริ่มต้นของโปรแกรม คือ ตำแหน่งที่โปรแกรมจะเริ่มเก็บในหน่วยความจำหลัก คำนวณได้ดังนี้

ตำแหน่ง = 5EF8H +(n * [B2H + (OCBAH)]) + ๑

เมื่อ n คือ จำนวนแฟ้มข้อมูลที่สามารถใช้ได้พร้อมกันในโปรแกรม

5EF8H คือ ตำแหน่งเริ่มต้นของแฟ้มข้อมูลข้อมูลที่ ๑

B2H คือ ค่า offset ของแต่ละแฟ้มข้อมูล

(OCBAH) คือ ค่าความยาวสูงสุดของระเบียน

เช่น ถ้าสามารถเปิดแฟ้มข้อมูลพร้อมกันได้ ๓ แฟ้ม และมีความยาวสูงสุดของระเบียนเป็น ๑๒๘ ไบต์ ดังนั้นตำแน่งเริ่มต้นของโปรแกรมจะอยู่ที่ตำแหน่ง 628FH โดยคำนวณได้จาก 5EF8H + (๓ * (00B2H + ๑๒๘)) + ๑ เป็นต้น

๒.๑๐ การแบ่งหน่วยความจำหลักของอินเตอร์เพรตเตอร์

(Memory Map of MBASIC)

เมื่ออินเตอร์เพรตเตอร์ถูกโหลดลงลงหน่วยความจำหลักเพื่อใช้งานแล้ว มันจะจัดแบ่งหน่วยความจำหลักตามการใช้งาน ซึ่งจากที่อธิบายมาตั้งแต่ต้น จะแสดงการแบ่งหน่วยความจำหลักของอินเตอร์เพรตเตอร์ด้วยแผนภาพ ดังรูปที่ ๒.๕