บทที่ ๒

อินเตอร์พริเตอร์ภาษาเบสิค

อินเตอร์พริเตอร์ เป็นตัวแปลภาษาแบบหนึ่ง ซึ่งในการแปลจะทำการแปลทีละบรรทัด หรือคำสั่ง และปฏิบัติการโดยทันทีจนกว่าจะหมดคำสั่งในโปรแกรมนั้น แตกต่างจากคอมไพเลอร์ (Compiler) ซึ่งจะต้องแปลคำสั่งในโปรแกรมหมดทั้งโปรแกรมเป็นภาษาเครื่องเสียก่อน แล้วจึงปฏิบัติการทำงานตามคำสั่งจากภาษาเครื่องของโปรแกรมนั้น ซึ่งเป็นข้อแตกต่างที่เห็นได้ชัดมาก นอกจากนี้ การใช้ภาษาที่ใช้อินเตอร์พริเตอร์ ยังสามารถแก้ไขเปลี่ยนแปลงโปรแกรมได้ง่าย ซึ่งประหยัดเวลากว่าคอมไพเลอร์ที่จะต้องทำการแปลโปรแกรมหมดทั้งโปรแกรมเสียก่อน จึงจะทราบความผิดพลาดของโปรแกรมนั้น ๆ

๒.๑ โมดการทำงาน [๑] (Mode of operation)

โปรแกรมที่ต้องการทำงานจะต้องถูกนำเข้ามาในหน่วยความจำหมดทั้งโปรแกรม อินเตอร์พริเตอร์จะอ่านคำสั่งในโปรแกรมในแต่ละบรรทัด มาทำการแปลว่าเป็นคำสั่งใด แล้วจึงข้ามไปยังอินเตอร์พริเตอร์ส่วนที่ทำงานของคำสั่งนั้น จากนั้นจึงอ่านคำสั่งต่อไป เพื่อทำงานไปเรื่อย ๆ จนหมดโปรแกรม โมดการทำงานของอินเตอร์พริเตอร์แบ่งออกเป็น ๒ โมดดังนี้คือ

๒.๑.๑ ไดเรคโมด (Direct mode) เป็นโมดที่สามารถรับคำสั่งและปฏิบัติตามคำสั่งนั้นได้ทันที นอกจากนี้ ยังสามารถรับคำสั่งได้หลายคำสั่ง แต่จะต้องอยู่ในบรรทัดเดียวกัน (เรียกว่า มัลติ-สเตทเมนท์ (Multi-statement)) บางคำสั่งถูกบังคับให้สามารถใช้ได้กับไดเรคโมดเท่านั้น เช่น LIST NEW RUN เป็นต้น

---

[๑] Nippon Electric Co., Ltd., PC-8001B N-Basic Reference Manual (Nippon Electric Co., Ltd. 1981), p. 1-1.

๒.๑.๒ อินไดเรคโมด (Indirect mode) เป็นโมดที่รับคำสั่งที่ต้องการมีเลขหมายบรรทัด (Line number) อยู่ประจำบรรทัดของกลุ่มคำสั่งนั้น อินเตอร์พรีเตอร์จะนำคำสั่งเหล่านี้ไปไว้ในหน่วยความจำในส่วนของโปรแกรม โดยยังไม่มีการปฏิบัติการทำงาน จนกว่าจะได้รับคำสั่ง RUN

๒.๒ การแทนข้อมูลในหน่วยความจำ

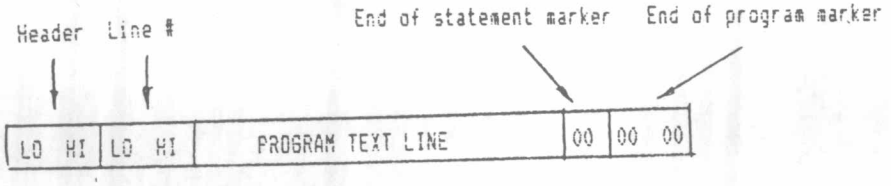
เพื่อเป็นการประหยัดเนื้อที่ในหน่วยความจำ การเก็บข้อมูลหรือโปรแกรมจึงต้องมีการจัดรูปแบบให้เหมาะสม แบ่งออกดังนี้คือ

๒.๒.๑ คำสั่ง (Statements)

คำสั่งที่ใช้ในภาษาเบสิค เนื่องจากเป็นภาษาชั้นสูง จึงประกอบด้วยตัวอักษรหลายตัวเพื่อให้อ่านแล้วได้ใจความ เช่น PRINT GOTO GOSUB เป็นต้น การที่จะนำตัวอักษรเหล่านี้มาเก็บในหน่วยความจำ จะเป็นการสิ้นเปลืองเนื้อที่โดยไม่จำเป็น จึงควรแปลงคำสั่งเหล่านี้เป็นรหัส เพื่อลดจำนวนตัวอักษร ระหัสนี้เรียกว่า โทเกน (Token) ดูตัวอย่าง โทเกนที่ภาคผนวก ก.

๒.๒.๒ รูปแบบบรรทัด (Line format)

โปรแกรมเมื่ออยู่ในไดเรคโมด จะถูกนำมาเก็บในหน่วยความจำ ซึ่งการเก็บจะมีรูปแบบเพื่อสะดวกในการทำงานของอินเตอร์พรีเตอร์ (ดูรูปที่ ๒.๑)



รูปที่ ๒.๑ รูปแบบบรรทัด

การจัดเก็บบรรทัดของโปรแกรม (Program text) ในหน่วยความจำ จะจัดเรียงกันตามหมายเลขบรรทัด จากน้อยไปหามาก ติดต่อกันทั้งโปรแกรม ดังมีรายละเอียดดังนี้

๒.๒.๒.๑ หัวบรรทัดแบ่งออกเป็น ๒ ไบท์ ซึ่งจะเก็บแอดเดรสของบรรทัดต่อไป (Next line address) โดยไบท์ที่ ๑ เป็นไบท์ที่มีลำดับตำแหน่งต่ำ (Low order byte) ไบท์ที่ ๒ เป็นไบท์ที่มีลำดับตำแหน่งสูง (High order byte) การจัดเก็บเป็นเลขฐานสิบหก

๒.๒.๒.๒ หมายเลขบรรทัด แบ่งออกเป็น ๒ ไบท์ ซึ่งจะเก็บหมายเลขบรรทัด โดยไบท์ที่ ๑ เป็นไบท์ที่มีลำดับตำแหน่งต่ำ ไบท์ ที่ ๒ เป็นไบท์ที่มีลำดับตำแหน่งสูง การจัดเก็บเป็นเลขฐานสิบหก

๒.๒.๒.๓ บรรทัดของโปรแกรม จะจัดเก็บคำสั่งที่ถูกแปลงเป็นโทเกนเรียบร้อยแล้ว รวมทั้งตัวอักษรที่ประกอบกับคำสั่งนั้น ๆ

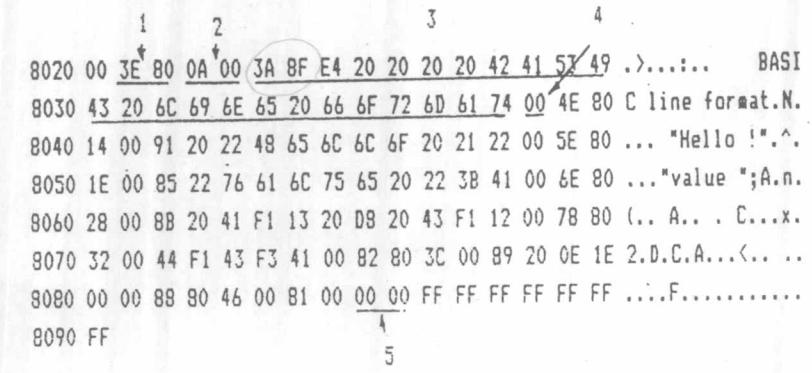
๒.๒.๒.๔ เครื่องหมายแสดงการหมดบรรทัด ซึ่งใช้เลขศูนย์จำนวน ๑ ไบท์

๒.๒.๒.๕ เครื่องหมายแสดงการหมดโปรแกรม ซึ่งใช้เลขศูนย์ จำนวน ๒ ไบท์

```
10 '    BASIC line format
20 PRINT "Hello !"
30 INPUT"value ";A
40 IF A=2 THEN C=1
50 D=C+A
60 GOTO 30
70 END
```

รูปที่ ๒.๒ โปรแกรม ที่ต้องการเก็บไว้ในหน่วยความจำ

BASIC LINE FORMAT REPRESENTATION IN MEMORY



รูปที่ ๒.๓ แสดงการเก็บโปรแกรมรูปที่ ๒.๒ ในหน่วยความจำ

จากรูปที่ ๒.๓ จะมีหมายเลขแสดงดังนี้

๑) หัวบรรทัดของหมายเลขบรรทัด ๑๐ เก็บแอดเดรสของหมายเลขบรรทัด ๒๐ คือ 803EH

๒) หมายเลขบรรทัดจะเก็บ 00 0AH ซึ่งเป็นเลขฐานสิบหก (เลข ๑๐ ฐานสิบ)

๓) บรรทัดของโปรแกรม คือข้อความต่าง ๆ ภายใน หมายเลขบรรทัด ๑๐

๔) เครื่องหมายแสดงการหมดบรรทัด

๕) เครื่องหมายแสดงหมดโปรแกรม

๒.๒.๓ ค่าคงที่ (Constants)

ในการเขียนโปรแกรมย่อมมีการกำหนดค่าให้กับตัวแปร ฉะนั้น การเก็บค่าต่าง ๆ จึงมีรูปแบบแตกต่างกันออกไป ดังนี้

๒.๒.๓.๑ ค่าคงที่ชนิดสตริง (String constants) ใช้กำหนดตัวอักษรหรือตัวเลขที่ไม่สามารถนำไปคำนวณให้กับตัวแปรชนิดนี้ แต่ละตัวอักษรหรือตัวเลขนี้ใช้แทนด้วยรหัสแอสคี (ASCII) และใช้เนื้อที่ในหน่วยความจำ จำนวน ๑ ไบท์แทน ๑ ตัวอักษร (ดูรูป ๒.๔)

```
10 N$="STRING Constants representation in memory"
```

```
8020 00 54 80 0A 00 4E 24 F1 22 53 54 52 49 4E 47 20 .T...N$.*STRING
8030 43 6F 6E 73 74 61 6E 74 73 20 72 65 70 72 65 73 Constants repres
8040 65 6E 74 61 74 69 6F 6E 20 69 6E 20 6D 65 6D 6F entation in memo
8050 72 79 22 00 00 00 00 00 00 00 00 00 00 00 00 00 ry"..............
8060 00
```

รูปที่ ๒.๔ แสดงการเก็บค่าคงที่ที่กำหนดให้กับตัวแปร N$ ในหน่วยความจำ

๒.๒.๓.๒ ค่าคงที่ชนิดจำนวนเต็ม (Integer constants) เป็นเลขจำนวนเต็มที่อยู่ระหว่าง -๓๒๗๖๘ กับ + ๓๒๗๖๗ การเก็บแบ่งเป็น ๓ ลักษณะ ดังนี้

๑. เลขจำนวนเต็มที่มีค่า ๐ ถึง ๙ จะใช้เนื้อที่ในการเก็บค่า ๑ ไบท์ เครื่องหมายอีก ๑ ไบท์ (ถ้ามี) โดย

๑๑	แทน เลข	๐
๑๒	"	๑
๑๓	"	๒
๑๔	"	๓
๑๕	"	๔
๑๖	"	๕
๑๗	"	๖
๑๘	"	๗
๑๙	"	๘
1A	"	๙

F3 แทน เครื่องหมาย บวก

F4 " ลบ

```
10 A=1

8020 00 29 80 0A 00 41 F1 12 00 00 00 00 00 00 00 00 .)...A.?.........
8030 00
       10 A=-1

8020 00 2A 80 0A 00 41 F1 F4 12 00 00 00 0A 00 00 00 .*...A..........
8030 00
```

รูปที่ ๒.๕ แสดงการเก็บค่าคงที่ชนิดจำนวนเต็มแบบที่ ๑ ในหน่วยความจำ

๒. เลขจำนวนเต็มที่มีค่าตั้งแต่ ๑๐ ถึง ๒๕๕ จะใช้เนื้อที่ในการเก็บค่า ๑ ไบท์ จัดเก็บเป็นเลขฐานสิบหก โดยมีแฟลก (Flag) ซึ่งแสดงด้วย OFH จำนวน ๑ ไบท์ และเครื่องหมายอีก ๑ ไบท์ (ถ้ามี) รวมไม่เกิน ๓ ไบท์

```
10 B=100

8020 00 2A 80 0A 00 42 F1 0F 64 00 00 00 0A 00 00 00 .*...B..d.......
8030 00
```

รูปที่ ๒.๖ แสดงการเก็บค่าคงที่ชนิดจำนวนเต็มแบบที่ ๒ ในหน่วยความจำ

๓. เลขจำนวนเต็มที่มีค่าตั้งแต่ ๒๕๕ ถึง ๓๒๗๖๗ จะใช้เนื้อที่ในการเก็บค่า ๒ ไบท์ จัดเก็บเป็นเลขฐานสิบหก โดยมีแฟลก ซึ่งแสดงด้วย 1CHจำนวน ๑ ไบท์ และเครื่องหมาย ๑ ไบท์ (ถ้ามี) รวมไม่เกิน ๔ ไบท์

```
10 C=500

8020 00 2B 80 0A 00 43 F1 1C 2C 01 00 00 00 0A 0A 43 .+...C..,......C
8030 00
```

รูปที่ ๒.๗ แสดงการเก็บค่าคงที่ชนิดจำนวนเต็ม แบบที่ ๓ ในหน่วยความจำ

๒.๒.๓.๓ ค่าคงที่ชนิดทศนิยม (Single precision constants) เป็นเลขทศนิยม ตั้งแต่ ๐.๐ ถึง 1E + 38 ใช้เนื้อที่ในการเก็บค่า ๔ ไบท์ โดยมีแฟลก ซึ่งแสดงด้วย 1DH จำนวน ๑ ไบท์ และเครื่องหมายอีก ๑ ไบท์ (ถ้ามี) รวมไม่เกิน ๖ ไบท์

การจัดเก็บค่าจะเก็บค่าในรูปเลขยกกำลัง โดยภายใน ๔ ไบท์ ไบท์สุดท้ายเก็บกำลัง (ดูรูป ๒.๘)

```
10 D=1E+07

8020 00 2D 80 0A 00 44 F1 1D 7F 96 18 98 00 00 00 0A .-...D.........
8030 00
```

รูปที่ ๒.๘ แสดงการเก็บค่าคงที่ชนิดทศนิยม

ค่าคงที่ชนิดนี้ โดยปกติถ้าไม่ใส่จุดทศนิยม หรือตัวตาม (Suffix) จะใช้เก็บตัวเลขตั้งแต่ ๓๒๗๖๗ ถึง ๙๙๙๙๙๙

๒๗๒๗๓๐๔ ค่าคงที่ชนิดเลขทศนิยม ดับเบิลพรีซิชัน (Double precision constants) เป็นเลขทศนิยม ตั้งแต่ 0.0 ถึง 1D + 38 แต่เนื่องจากสามารถแสดงจำนวนเต็มหน้าจุดทศนิยมได้ถึง ๑๖ หลัก จึงใช้อีก D แทน E ซึ่งแสดงได้เพียง ๖ หลัก ถ้าต้องการใช้คงที่ชนิดนี้บางครั้งอาจต้องตามด้วย # หลักตัวคงที่นั้น ๆ เพื่อให้สามารถแสดงเลขหน้าจุดทศนิยมมากกว่า ๖ หลัก ค่าคงที่ชนิดนี้ ใช้เนื้อที่ในการเก็บค่า ๘ ไบท์ โดยมีแฟลก ซึ่งแสดงด้วย 1FH จำนวน ๑ ไบท์ และเครื่องหมายอีก ๑ ไบท์ (ถ้ามี) รวมไม่เกิน ๑๐ ไบท์

การจัดเก็บค่ามีลักษณะคล้ายกับค่าคงที่ชนิดทศนิยม เพียงแต่จำนวนไบท์ในการเก็บตัวเลขจำนวนเต็มมากกว่าค่าคงที่ชนิดทศนิยม ๔ ไบท์ ฉะนั้น สามารถเก็บค่าได้ละเอียดขึ้น แต่การยกกำลังยังสูงสุดเท่าเดิม คือไม่เกิน $๑๐^{+๓๘}$ (ดูรูป ๒.๙)

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

```
10 D=5.555555556D+28

8020 00 31 80 0A 00 44 F1 1F 38 ED 96 7D 7A 82 33 E0 .1...D..8..}z.3.
8030 00
```

รูปที่ ๒.๙ แสดงการเก็บค่าคงที่ชนิดทศนิยม

๒.๒.๔ ตัวแปร (Variables)

ในการตั้งชื่อตัวแปรนี้ สามารถตั้งความยาวเท่าไรก็ได้ แต่มีนัยสำคัญเพียง ๒ ตัวแรกเท่านั้น ตัวแรกจะต้องเป็นตัวอักษรเท่านั้น ที่เหลืออาจเป็นตัวเลขหรือตัวอักษรก็ได้ นอกจากนั้นการตั้งชื่อจะต้องไม่ตรงกับคำเฉพาะ (Reserved words) ที่ใช้ในภาษาเบสิค (ดูภาคผนวก ก.)

ชนิดของตัวแปรที่ใช้เก็บค่าแบ่งออกเป็น ๔ ชนิดคือ

๒.๒.๔.๑ ตัวแปรสตริง (String variable) ตัวแปรชนิดนี้จะมีตัวตามเป็นเครื่องหมายดอลล่าร์ ($) ซึ่งใช้เก็บตัวอักษรหรือตัวเลขก็ได้ แต่จะต้องอยู่ระหว่างเครื่องหมายอัญประกาศ เช่น AN$, C1$, ABC$ เป็นต้น

การเก็บภายในหน่วยความจำนั้น ตัวแปรจะถูกเก็บไว้ท้ายโปรแกรม จะใช้เนื้อที่ ๖ ไบท์ ต่อตัวแปรหนึ่ง ๆ ดังนี้ (ดูรูปที่ ๒.๑๐)

๒.๒.๔.๑.๑ ไบท์แรก จะเป็นไบท์ที่บอกชนิดของตัวแปร แทนด้วยรหัส ๐๓

๒.๒.๔.๑.๒ ไบท์ต่อมาจะเก็บชื่อตัวที่ ๒ ของตัวแปร และอีก ๑ ไบท์ จะเก็บเป็นชื่อตัวแรกของตัวแปรนั้น การที่เก็บชื่อตัวที่ ๒ ของตัวแปรก่อน เพื่อเพิ่มประสิทธิภาพและความเร็วในการค้นหา เพราะว่าส่วนใหญ่เวลาตั้งชื่อตัวแปรจะเป็นดังนี้ A1$, A2$ เป็นต้น

๒.๒.๔.๑.๓ ไบท์ที่ ๔ ใช้เก็บความยาวของตัวอักษรที่เก็บภายในสตริงสแตค (String stack) ซึ่งเป็นส่วนของหน่วยความจำที่ใช้เก็บตัวอักษร หรือตัวเลขของตัวแปรชนิดนี้

๒.๒.๔.๑.๔ ไบท์ที่ ๕ และ ๖ ใช้เก็บแอดเดรสที่ชี้ไปยังสตริงสแตค ตัวอักษรที่เก็บในสตริงสแตคนั้น จะเก็บเป็นรหัสแอสกี

```
10 B$="Test"
20 INPUT"STRING TEST ";G$
30 END
```

```
8020 00 2F 80 0A 00 42 24 F1 22 54 65 73 74 22 00 45 ./...B$."Test".E
8030 80 14 00 85 22 53 54 52 49 4E 47 20 54 45 53 54 ...."STRING TEST
8040 22 3B 47 24 00 4B 80 1E 00 81 00 00 00 03 00 42 ";G$.K.........B
8050 04 29 80 03 00 47 17 E9 E9 00 00 00 00 00 00 00 .)...G..........

                    STRING STACK

E9E0 00 00 00 00 00 00 00 00 00 48 65 6C 6C 6F 20 53 .........Hello S
E9F0 54 52 49 4E 47 20 53 54 41 43 4B 20 74 65 73 74 TRING STACK test
```

รูปที่ ๒.๑๐ แสดงการเก็บตัวแปรชนิดสตริง ในหน่วยความจำ

๒.๒.๔.๒ ตัวแปรชนิดจำนวนเต็ม (Integer variables) ตัวแปรชนิดนี้ใช้เก็บเลขจำนวนเต็มที่อยู่ระหว่าง -๓๒๗๖๘ ถึง + ๓๒๗๖๗ ตัวแปรที่แสดงว่าเป็นตัวแปรจำนวนเต็ม จะมีตัวตามเป็นเครื่องหมายเปอร์เซนต์ (%) เช่น AB %, MA %, C2 % เป็นต้น

๒.๒.๔.๒.๑ ไบท์แรกจะเป็นไบท์ที่บอกชนิดของตัวแปรแทนด้วยรหัส ๐๒

๒.๒.๔.๒.๒ ไบท์ต่อมาจะเก็บชื่อตัวที่ ๒ ของตัวแปร และอีก ๑ ไบท์จะเก็บเป็นชื่อตัวแรกของตัวแปรนั้น

๒.๒.๔.๒.๓ ไบท์ที่ ๔ และ ๕ ใช้เก็บค่าของตัวแปร โดยเก็บเป็นเลขฐานสิบหก

```
10 AB%=-2
20 CD%=200
30 EF%=5000

8020 00 2C 80 0A 00 41 42 25 F1 F4 13 00 37 80 14 00 .,...AB%....7...
8030 43 44 25 F1 0F C8 00 43 80 1E 00 45 46 25 F1 1C CD%....C...EF%..
8040 88 13 00 00 00 02 42 41 FE FF 02 44 43 C8 00 02 ......BA...DC...
8050 46 45 88 13 02 00 00 00 00 00 00 00 00 00 00 FE..............
8060 00
```

รูปที่ ๒.๑๑ แสดงการเก็บตัวแปรชนิดจำนวนเต็มในหน่วยความจำ

007342

๒.๒.๔.๓ ตัวแปรชนิดเลขทศนิยม (Single precision voriables) ตัวแปรชนิดนี้ใช้เก็บตัวเลขที่มีจำนวนระหว่าง 1.0E-38 ถึง 1.0E + 38 ตัวแปรชนิดนี้ไม่จำเป็นจะต้องมีตัวตาม แต่ถ้าจะระบุก็ไว้เครื่องหมายอัศเจรีย์ (!) เช่น AB, CD !, M ! เป็นต้น

การเก็บภายในหน่วยความจำนั้น ตัวแปรจะถูกเก็บไว้ท้ายโปรแกรม จะใช้เนื้อที่ ๗ ไบท์ต่อตัวแปรหนึ่ง ๆ ดังนี้ (รูปที่ ๒.๑๒)

๒.๒.๔.๓.๑ ไบท์แรกจะเป็นไบท์ที่บอกชนิดของตัวแปร แทนด้วยรหัส ๐๔

๒.๒.๔.๓.๒ ไบท์ต่อมาจะเก็บชื่อตัวที่ ๒ ของตัวแปร และอีก ๑ ไบท์ จะเก็บเป็นชื่อตัวแรกของตัวแปรนั้น

๒.๒.๔.๓.๓ ไบท์ที่ ๔ ถึง ๗ ใช้เก็บค่าของตัวแปร โดยไบท์ที่ ๗ ใช้เก็บเลขยกกำลังของสอง ไบท์ที่ ๖, ๕ และ ๔ ใช้เก็บค่าเป็นเลขสิบหก ซึ่งมีนัยสำคัญของแต่ละไบท์ตามลำดับ

```
10 AB!=-2
20 CD!=200
30 EF!=5000

8020 00 2C 80 0A 00 41 42 21 F1 F4 13 00 37 80 14 00 .,...AB!....7...
8030 43 44 21 F1 0F C8 00 43 80 1E 00 45 46 21 F1 1C CD!....C...EF!..
8040 88 13 00 00 00 04 42 41 00 00 80 82 04 44 43 00 ......BA.....DC.
8050 00 48 88 04 46 45 00 40 1C 8D 1E 00 00 00 00 00 .H..FE.@........
8060 00
```

รูปที่ ๒.๑๒ แสดงการเก็บตัวแปรชนิดเลขทศนิยมในหน่วยความจำ

๒.๒.๔.๔ ตัวแปรชนิดเลขทศนิยมดับเบิลพรีซิชัน (Double precision variables) ตัวแปรชนิดนี้ใช้เก็บตัวเลขที่มีจำนวนเท่ากับ ตัวแปรชนิดทศนิยม แต่สามารถเก็บตัวเลขได้ละเอียดกว่าตัวแปรชนิดทศนิยม ตัวแปรที่แสดงว่าเป็นตัวแปรจำนวนเต็ม จะมีตัวตามเป็นเครื่องหมาย # เช่น AB #, CD #, M # เป็นต้น

การเก็บภายในหน่วยความจำนั้น ตัวแปรจะถูกเก็บไว้ท้ายโปรแกรม จะใช้เนื้อที่ ๑๑ ไบท์ ต่อตัวแปรหนึ่ง ๆ ดังนี้ (รูป ๒.๑๓)

๒.๒.๔.๔.๑ ไบท์แรกจะเป็นไบท์ที่บอกชนิดของตัวแปร แทนด้วยรหัส ๐๘

๒.๒.๔.๔.๒ ไบท์ต่อมาจะเก็บชื่อตัวที่ ๒ ของตัวแปร และอีก ๑ ไบท์จะเก็บเป็นชื่อตัวแรกของตัวแปรนั้น

๒.๒.๔.๔.๓ ไบท์ ๔ ถึง ๑๑ ใช้เก็บค่าตัวแปร โดยไบท์ที่ ๑๑ ใช้เก็บเลขยกกำลังของสอง ไบท์ที่ ๑๐ ถึง ๔ ใช้เก็บค่าเป็นเลขฐานสิบหก ซึ่งมีนัยสำคัญของแต่ละไบท์ตามลำดับ

```
10 AB#=-2
20 CD#=200
30 EF#=5000

8020 00 2C 80 0A 00 41 42 23 F1 F4 13 00 37 80 14 00 .,...AB#....7...
8030 43 44 23 F1 0F C8 00 43 80 1E 00 45 46 23 F1 1C CD#....C...EF#..
8040 88 13 00 00 00 08 42 41 00 00 00 00 00 00 80 82 ......BA........
8050 08 44 43 00 00 00 00 00 00 48 88 08 46 45 00 00 .DC......H..FE..
8060 00 00 00 40 1C 80 08 00 00 00 00 00 00 00 00 00 ...@............
8070 00
```

รูปที่ ๒.๑๓ แสดงการเก็บตัวแปรชนิดเลขทศนิยมดับเบิลพรีซิชั่นในหน่วยความจำ

๒.๒.๕ ตัวแปรหมวด (Array)

ตัวแปรหมวด สามารถสร้างได้โดยคำสั่ง DIM และสามารถกำหนดจำนวนมิติได้มากที่สุด ๒๕๕ มิติ ซึ่งแต่ละมิติมีค่าได้ระหว่าง ๐ ถึง ๖๕๕๓๕

ถ้ามีการใช้ตัวแปรหมวดภายในโปรแกรมโดยไม่ใช้คำสั่งไว้ล่วงหน้า กรณีนี้สามารถใช้ได้เพียงมิติเดียว และมีค่าระหว่าง ๐ ถึง ๑๐

การใช้ตัวแปรหมวด สามารถกำหนดให้เป็นตัวแปรชนิดใดชนิดหนึ่งก็ได้ การเก็บภายในหน่วยความจำนั้น ตัวแปรหมวดจะถูกเก็บไว้ท้ายโปรแกรม และหลังจากเนื้อที่การเก็บตัวแปรปกติ จะใช้เนื้อที่ในการเก็บดังนี้ (ดูรูป ๒.๑๔)

๒.๒.๕.๑ ไบท์แรกใช้เก็บชนิดของตัวแปรนั้น ๆ

๒.๒.๕.๒ ไบท์ต่อมาใช้เก็บชื่อตัวที่ ๒ ของตัวแปร และอีก ๑ ไบท์ใช้เก็บเป็นชื่อตัวแรกของตัวแปรนั้น

๒.๒.๕.๓ ไบท์ที่ ๔ และ ๕ ใช้เก็บความยาวของตัวแปรหมวด ซึ่งได้จาก

จำนวนไบท์ในการเก็บค่าของตัวแปรชนิดนั้น ๆ + จำนวนมิติ x2 + 1

๒.๒.๕.๔ ไบท์ที่ ๖ ใช้เก็บจำนวนมิติ

๒.๒.๕.๕ ไบท์ต่อ ๆ ไป ใช้เก็บค่าของมิติ ๆ ละ ๒ ไบท์

๒.๒.๕.๖ ไบท์ต่อจากใช้เก็บค่าของมิติจะใช้เก็บค่าของตัวแปรหมวดจำนวนไบท์ ขึ้นกับชนิดของตัวแปร

```
10 DIM D$(1,2,3)
20 D$(1,1,2)="String array"

8020 00 31 80 0A 00 86 20 44 24 28 12 2C 13 2C 14 29 .1.... D$(.,.,.)
8030 00 4E 80 14 00 44 24 28 12 2C 12 2C 13 29 F1 22 .N...D$(.,.,.)."
8040 53 74 72 69 6E 67 20 61 72 72 61 79 22 00 00 00 String array"...
8050 03 00 44 4F 00 03 04 00 03 00 02 00 00 00 00 00 ..DO............
8060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
8070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
8080 00 00 00 00 00 00 00 00 0C 40 80 00 00 00 00 00 .........@......
8090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
80A0 00 00 00 00 FF FF FF FF FF FF FF FF FF FF FF FF ................
80B0 FF
10 DIM S%(1,2)
20 S%(0,1)=100

8020 00 2F 80 0A 00 86 20 53 25 28 12 2C 13 29 00 3E ./.... S%(.,.).>
8030 80 14 00 53 25 28 11 2C 12 29 F1 0F 64 00 00 00 ...S%(.,.)..d...
8040 02 00 53 11 00 02 03 00 02 00 00 00 00 00 64 00 ..S...........d.
8050 00 00 00 00 00 00 FF FF FF FF FF FF FF FF FF FF ................
8060 FF
20 N%(10)=100

8020 00 2F 80 14 00 4E 25 28 0F 0A 29 F1 0F 64 00 00 ./...N%(..)..d..
8030 00 02 00 4E 19 00 01 0B 00 00 00 00 00 00 00 00 ...N............
8040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 64 00 FF ..............d..
8050 FF
```

รูปที่ ๒.๑๔ แสดงการเก็บตัวแปรหมวดในหน่วยความจำ

๒.๓ การแบ่งหน่วยความจำ [๑] (Memory map)

การจัดการในหน่วยความจำ เพื่อแบ่งหน่วยความจำในการใช้งาน สามารถแบ่งออกเป็น ๖ ส่วนดังนี้ (ดูรูป ๒.๑๔)

๒.๓.๑ หน่วยความจำที่เก็บอินเตอร์พริเตอร์ภาษาเบสิค ซึ่งอาจเป็นรอม (ROM) หรืออีพรอม (EPROM-Erasable Programmable Read Only Memory) ก็ได้

๒.๓.๒ หน่วยความจำส่วนที่เก็บโปรแกรมภาษาเบสิค และตัวแปรต่าง ๆ

๒.๓.๓ หน่วยความจำที่ทำหน้าที่เป็น สแตค เพื่อการใช้งานของอินเตอร์พริเตอร์และโปรแกรมภาษาเบสิค เช่น GOSUB RETURN เป็นต้น

๒.๓.๔ หน่วยความจำส่วนที่เก็บค่าคงที่ชนิดสตริง (String area)

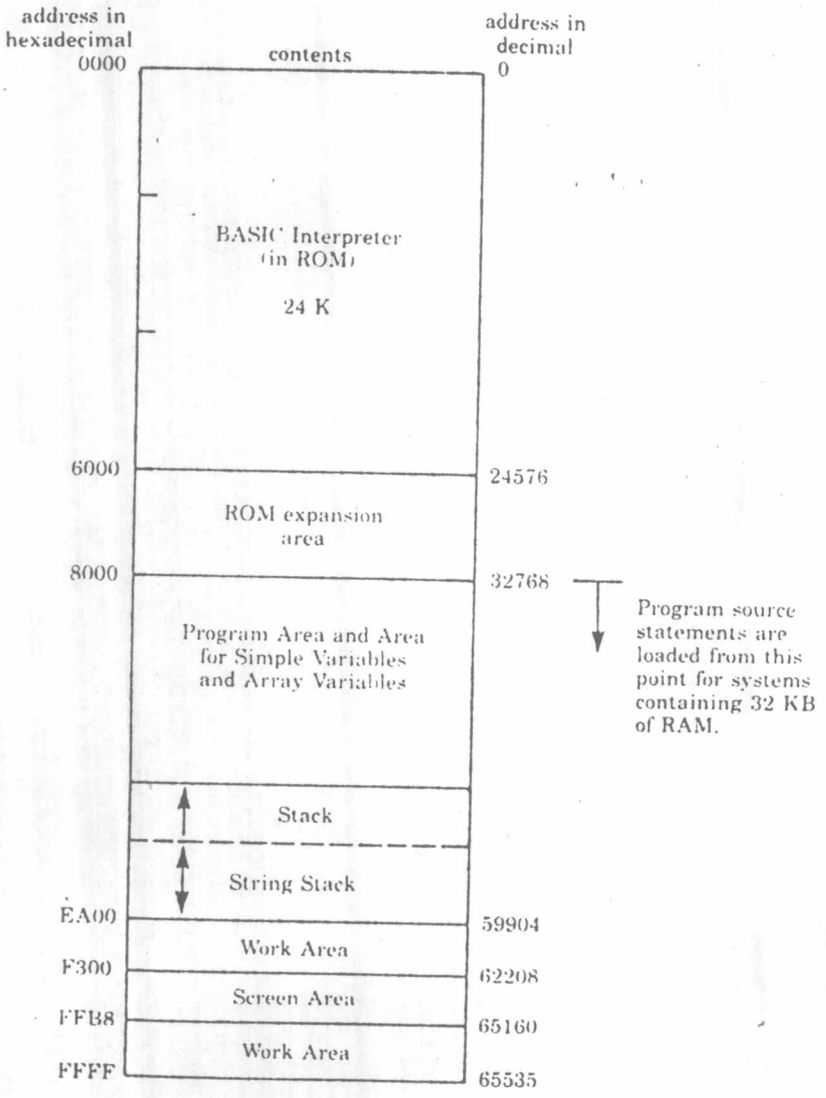
๒.๓.๕ หน่วยความจำส่วนที่ใช้งานทั่วไป (Work area)

๒.๓.๖ หน่วยความจำส่วนที่เก็บข้อมูลสำหรับแสดงผลบนจอภาพ (Screen area)

---

[๑] Nippon Electric Co., Ltd., PC-8001B N-Basic Reference Manual (Nippon Electric Co., Ltd. 1981), Appendix B.

MEMORY MAP



รูปที่ ๒.๑๔ แสดงการแบ่งหน่วยความจำของไมโครคอมพิวเตอร์ NEC PC-8000