

## บทที่ 6 : Record Input/Output

record I/O เป็นรูปแบบมาตรฐานของการประเมินผล file ในการเขียนโปรแกรมทางด้านธุรกิจและมีประสิทธิภาพมากกว่า Stream I/O, Record I/O อ่านข้อมูล (records) เข้าไปได้ ในเนื้อที่หน่วยความจำหลักแล้วพิมพ์ข้อมูล (records) ออกจากหน่วยความจำหลักทีละตัว character by character) ไม่มีการเปลี่ยนรูปของข้อมูลที่เป็นตัวเลข ซึ่งลักษณะคล้ายที่สุดของ Record I/O คือ Edit-directed stream I/O ซึ่งใช้โครงสร้างสำหรับ I/O areas.

ภายใต้ Record I/O เราต้อง declare file ด้วยคำสั่ง OPEN ก่อนการประมวลผล file นั้น และเมื่อเสร็จการประมวลผลต้องใช้คำสั่ง CLOSE file คำสั่ง READ และคำสั่ง WRITE ใช้อ่านข้อมูลเข้าเครื่องและนำข้อมูลออกจากเครื่องคอมพิวเตอร์, fields ที่เป็นตัวเลข เราเคย defined base ด้วยคำว่า DECIMAL และ BINARY ส่วน scale ใช้ keyword FIXED และ FLOAT แต่ใน Record I/O ถ้า input มาจากบัตร หรือ terminal จะใช้ format ดังกล่าวไม่ได้ format ดังกล่าวข้างต้นสำหรับข้อมูลที่เป็นตัวเลขใช้ได้เฉพาะ input ที่มาจาก tape หรือ disk เท่านั้น ในบทนี้จะครอบคลุมการใช้รูปแบบของ picture และ input ที่มูลค่าเป็นตัวเลขและ output เป็นมูลค่าที่มีการ edit

### 1) Record I/O file declare statements

ภายใต้ Record I/O file ทุก file ต้องใช้คำสั่ง declare เพื่อบอก compiler เกี่ยวกับชนิดของ device และคุณลักษณะอย่างอื่น ชื่อ file เป็น external identifier และความยาวของชื่อประกอบด้วย characters มากที่สุดไม่เกิน 7 ตัว (สำหรับ PL/C ไม่เกิน 6 ตัว)

**Input file** (IBM OS และ DOS Optimizing compilers) เฉพาะ card reader

IBM OS มีรูปแบบของคำสั่ง defines an input file ดังนี้

```
DCL file--name FILE RECORD INPUT ENV (F BLKSIZE (80));
```

IBM DOS มีรูปแบบดังนี้

```
DCL filename FILE RECORD INPUT ENV (MEDIUM (SYSIPT) F RECSILE (80)
BUFFERS (2));
```

## Printer files

character ตัวแรกของ a print record จะถูกสำรองไว้เป็น "control character" ภายใต้ Record I/O เช่นทำให้ printer เว้นบรรทัด หรือขึ้นกระดาษแผ่นใหม่ ถ้าความกว้างของกระดาษ เป็น 120 หรือ 132 character เราก้เขียน (121) หรือ (133) ตามลำดับ

IBM OS รูปแบบของคำสั่ง defines a printer file มีดังนี้

DCL filename FILE RECORD OUTPUT ENV (F BLKSIZE (133));

IBM DOS มีรูปแบบดังนี้

DCL filename FILE RECORD OUTPUT ENV (MEDIUM (SYSLST)F RECSIZE  
(121) BUFFERS (2) CTLASA);

## คำสั่ง OPEN

ก่อนอ่านข้อมูลหรือพิมพ์ข้อมูล ในโปรแกรมนั้นจะต้องมีคำสั่ง OPEN files

คำสั่ง OPEN หนึ่งคำสั่ง ใช้กับ file หนึ่ง file ดังนี้

OPEN FILE (filename-1);

OPEN FILE (filename-2);

หรือคำสั่ง OPEN หนึ่งคำสั่ง ใช้กับหลาย files ดังนี้

OPEN FILE (filename-1), FILE (filename-2),....;

คำสั่ง OPEN มีหน้าที่ดังต่อไปนี้

1) ตรวจสอบว่า files ที่มีชื่ออยู่นั้นมีอยู่จริงหรือไม่ (การตรวจสอบนี้มีความสำคัญพิเศษ สำหรับการประมวลผลข้อมูลใน tape และ disk)

2) ตรวจสอบว่า file นั้นพร้อมที่จะถูกนำมาใช้ได้หรือไม่ ภายใต้ระบบ multiprogramming เมื่อเครื่องคอมพิวเตอร์ execute โปรแกรมจำนวนหนึ่ง ตามลำดับ โปรแกรมอีกชุดหนึ่งอาจจะพร้อมที่จะประมวลผล โดยใช้ files นั้น, ถ้า files นั้นพร้อมที่จะถูกนำไปใช้ ระบบคอมพิวเตอร์ ก็จะจัดสรร file ให้กับโปรแกรม

## คำสั่ง CLOSE

เมื่อโปรแกรมอ่านหรือพิมพ์ records ทั้งหมดเสร็จเรียบร้อยแล้ว files เหล่านั้นต้อง CLOSE ให้หมด, คำสั่ง CLOSE หนึ่งคำสั่ง ใช้ CLOSE ได้หนึ่ง file มีรูปแบบดังนี้

CLOSE FILE (filename-1);

CLOSE FILE (filename-2);

หรือคำสั่ง CLOSE หนึ่งคำสั่ง CLOSE ได้หลาย files ดังนี้

CLOSE FILE (filename-1), FILE ( filename-2),....;

คำสั่ง CLOSE มีหน้าที่ดังนี้

1. พิมพ์ข้อมูล (records) ที่อยู่ในหน่วยความจำหลัก (ใน buffer) แล้วยังไม่ได้พิมพ์
2. ทำให้ file วางสำหรับโปรแกรมอื่นในระบบนั้น

### **SYSPRINT**

ในคอมพิวเตอร์บางระบบ ข้อความผิดพลาด สำหรับการ execute จะปรากฏทาง printer ก็ต่อเมื่อมีการ declare file พิเศษที่ชื่อว่า SYSPRINT

explicit declaration ใช้คำสั่ง DCL ดังนี้

```
DCL SYSPRINT FILE PRINT;
```

ส่วน implicit declaration ใช้คำสั่ง, ดังนี้

```
OPEN FILE (SYSPRINT);
```

### **คำสั่ง READ**

รูปแบบพื้นฐานสำหรับการอ่าน input data ภายใต Record I/O คือ

```
READ FILE (filename) INTO (inputarea);
```

เมื่อ filename เป็นชื่อของ input file ที่ declare แล้ว inputarea เป็นชื่อของ an element variable หรือ structure สำรองเนื้อที่สำหรับ input records เนื้อที่นี้ต้องมีขนาดเดียวกับ RECSIZE หรือ BLKSIZE ใน file ที่ declare แล้ว คำสั่ง READ จะ copy an input record ทีละ character โดยตรงไว้ใน an input area โดยลบมูลค่าเก่าทั้งหมด, คำสั่ง READ (ไม่เหมือนคำสั่ง GET) ต้องอ่านข้อมูลที่ละ record

### **คำสั่ง WRITE**

รูปแบบพื้นฐานสำหรับการ write data ภายใต Record I/O คือ

```
WRITE FILE (filename) FROM (outputarea);
```

เมื่อ filename เป็นชื่อของ output file ที่มีการ declare แล้ว

Output area เป็นชื่อของ an element variable หรือ structure สำรองสำหรับ a print record เนื้อที่นี้ต้องมีขนาดเดียวกับ RECSIZE หรือ BLKSIZE ใน file ที่ declare แล้ว ปกติขนาด 121 หรือ 133 เมื่อ control character อยู่ตำแหน่งซ้ายมือสุดของเนื้อที่นั้น

คำสั่ง WRITE copy print record, ทีละ character โดยตรงไปยัง print file แต่คำสั่ง WRITE จะไม่ clear print area อัตโนมัติ

### **CTLASA forms control**

เมื่อมีการพิมพ์ a record โปรแกรมเมอร์จะต้องบอก system ว่าต้องการเว้น 1, 2 หรือ 3 บรรทัดก่อนพิมพ์ หรือขึ้นกระดาษแผ่นใหม่, character ตัวแรกของ print area ถูกสำรองไว้

สำหรับ control character เพื่อบอก system ว่าต้องการจะให้ทำ action ใด ตารางข้างล่างนี้ เป็น option การใช้ ASA (ASA = American Standards Association)

ASA CHAR	ACTION
+	ไม่เว้นบรรทัดก่อนพิมพ์
b (blank)	เว้น 1 บรรทัด แล้วพิมพ์
0	เว้น 2 บรรทัด แล้วพิมพ์
-	เว้น 3 บรรทัด แล้วพิมพ์
1	ขึ้นกระดาษแผ่นใหม่และพิมพ์

สำหรับ character ตัวอื่นนอกเหนือจากนี้ system จะถือราวกับว่าเป็น blank และจะเว้นให้หนึ่งบรรทัดก่อนพิมพ์ การใช้ ASA form control ให้เขียน CTLASA ใน ENV option ของ file ที่ declare หรือ CTL360 สำหรับเครื่องคอมพิวเตอร์ของ IBM

หรือเราอาจกำหนดมูลค่าเริ่มต้นให้กับ control character ดังนี้

DCL 1 PRINT,

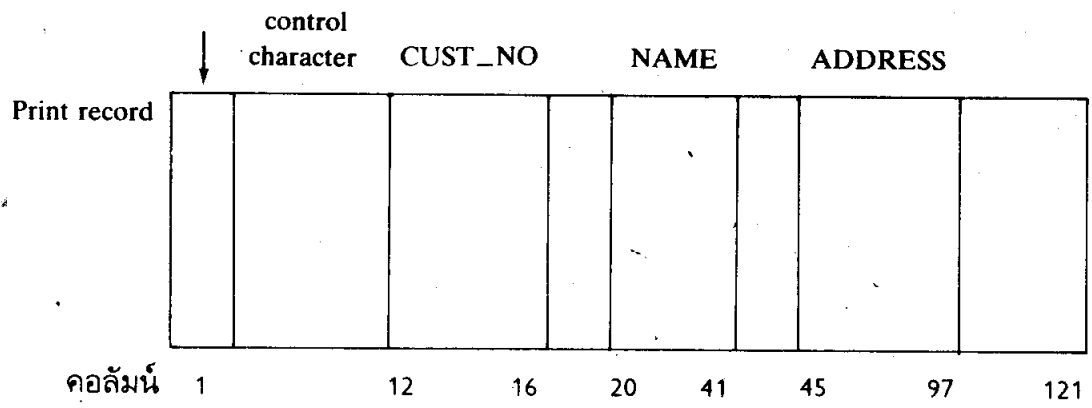
2 CTLCHR CHAR (1) INIT ('1')

หรือกำหนดมูลค่าให้กับ control character ดังนี้

CTLCHR = '1';

ตัวอย่างโปรแกรม พิมพ์รายละเอียดของลูกค้า ซึ่งมีชื่อและที่อยู่ โดยใช้ Record I/O

	CUST_NO	NAME	ADDRESS
Input record			
คอลัมน์	1	5 6	27 28 80



1 PROG 15 :

PROCEDURE OPTIONS (MAIN) :

```
2 DCL      FILEIN FILE RECORD INPUT ENV (MEDIUM SYSIPT)
           F RECSIZE (80) BUFFERS (2));
3 DCL      PRTR  FILE RECORD OUTPUT ENV (MEDIUM (SYSLST)
           F RECSIZE(121)BUFFERS(2) CTLASA) ;
4 DCL      SYSPRINT FILE PRINT ENV (BUFFERS(2).);
5 DCL      1 RECORDIN,
           3 CUSTNOIN      CHAR (05),
           3 NAMEIN        CHAR (22),
           3 ADDRIN        CHAR (53);
6 DCL      1 HEADING,
           3 CTLCHAR       CHAR (01) INIT ('1'),
           3 SPACE 0       CHAR (10) INIT (' '),
           3 TITLE 1       CHAR (05) INIT ('CUST#'),
           3 SPACE 1       CHAR (03) INIT (' '),
           3 TITLE 2       CHAR (22) INIT ('NAME'),
           3 SPACE 2       CHAR (03) INIT (' '),
           3 TITLE 3       CHAR (53) INIT ('ADDRESS'),
           3 SPACE 3       CHAR (24) INIT (' ');
7 DCL      1 CUST-LINE,
           3 SPACE 0       CHAR (10) INIT (' '),
           3 CTLCHAR       CHAR (01) INIT (' '),
           3 CUSTNOUT      CHAR (05),
           3 SPACE 1       CHAR (03) INIT (' '),
           3 NAMEOUT       CHAR (22),
           3 SPACE 2       CHAR (03) INIT (' '),
           3 ADDRROUT     CHAR (53),
           3 SPACE 3       CHAR (24) INIT (' ');
```

```

8 DCL      EOF          BIT (01) INIT ('0'B) ;
9          OPEN FILE (FILEIN), FILE (PRTR) ;
10         ON ENDFILE (FILEIN) EOF = '1'B;
11         WRITE FILE (PRTR) FROM (HEADING);/* PRINT HEADING */
12         READ FILE (FILEIN) INTO (RECORDIN) ;
13         DO WHILE (¬EOF);
14             CUSTNOUT = CUSTNOIN ;
15             NAMEOUT = NAMEIN ;
16             ADDROUT = ADDRIN ;
17             WRITE FILE (PRTR) FROM (CUST_LINE) ;
18             READ FILE (FILEIN) INTO (RECORDIN) ;
19         END ;
20         CLOSE FILE (FILEIN), FILE (PRTR) ;
21         SIGNAL FINISH ;
22     END PROG 15 ;

```

Output

CUST#	NAME	ADDRESS
10225	J SMITH	222 WEST 43RD AVENUE
25443	AP BROWN	4327 CHESTNUT DRIVE
30668	JA RICHARDSON	4157 LAKE BOULEVARD