บทที่ 10
STREAM ORIENTED INPUT/OUTPUT

แฟ้มข้อมูลชนิด STREAM ประกอบด้วย ชุด (sequence) ของตัวอักขระ ASCII คั่นด้วย linemarks และ pagemarks คำสั่ง STREAM I/O จัด facilities สำหรับการเข้าถึงข้อมูลตัวอักขระในแฟ้มข้อมูลชนิด STREAM โดยทั่วไป กฎข้างล่างนี้ใช้กับ STREAM I/O

1) ตำแหน่งคอลัมน์ สำหรับแฟ้มข้อมูลเริ่มด้วย 1

2) การเกิดแต่ละครั้งของ linemark หรือ pagemark เป็นการ reset ให้ตำแหน่งคอลัมน์เป็น 1

3) นอกเหนือจากนี้ ถ้าตัวอักขระ input หรือ output เป็น graphic ตำแหน่งคอลัมน์ จะเพิ่มขึ้นอีก 1

output, ถ้าตำแหน่งคอลัมน์ มากกว่า linesize, เครื่องจะพิมพ์ linemark ให้ เลขประจำบรรทัดจะเพิ่มขึ้นทีละ 1 และตำแหน่งคอลัมน์ จะถูก reset ให้เป็น 1

เมื่อเลขประจำบรรทัด มากกว่า ขนาดหน้ากระดาษ, เครื่องจะพิมพ์ pagemark ให้ แล้ว ตำแหน่งคอลัมน์ และเลขประจำบรรทัดจะถูก reset ให้เป็น 1

STREAM I/O ใน PL/I-80 แบ่งออกเป็น 3 รูปแบบ เรียกว่า list-directed, edit-directed และ line-directed

List-directed I/O ย้าย data items โดยไม่ต้องกำหนดรูปแบบข้อมูล

Edit-directed I/O กำหนด การเข้าถึง character data อย่างมีรูปแบบ

Line-directed I/O กำหนด การเข้าถึง character data ที่มีความยาวไม่คงที่ (variable length) ในรูปแบบที่ ไม่มีการอีดิท

หมายเหตุ line-directed I/O ใน PL/I-80 ประมวลผล (process) variable length AScII record โดยใช้คำสั่ง READ และคำสั่ง WRITE และเอาไปใช้ใน PL/I versions อื่น ๆ ไม่ได้

ชื่อต่อไปนี้ใช้ในรายละเอียดของคำสั่ง STREAM I/O

fname เป็น ไฟล์ ไอเดนติไฟเออร์

nl เป็น FIXED BINARY expression หมายถึงจำนวน linemarks ที่จะให้ข้ามไปใน input หรือจำนวน linemarks ที่จะให้พิมพ์ก่อน data item บน output

input-list เป็นรายชื่อตัวแปร แต่ละตัวให้แยกจากกันด้วยเครื่องหมาย comma หมายถึง data items ซึ่งจะถูกส่งค่าจาก input stream, input-list จะเป็นตัวบอกจำนวนและลำดับที่ของตัวแปรซึ่งกำหนดค่าโดย input data ใน stream ใน PL/I-80 ตัวแปรเหล่านี้ ต้องเป็นค่า scalar

ใน input-list อาจจะมี iterative DO loops รูปแบบของคำสั่ง DO ได้กล่าวถึงมาแล้ว ส่วน REPEAT clause เอามาใช้ด้วยไม่ได้

รูปแบบทั่วไป

(item-1, ..., item-n DO iteration)

ตัวอย่าง

```
GET LIST ((A(I),B(I) DO I = 1 TO 10));
```

output--list เป็นรายชื่อของ output items ประกอบด้วย ตัวแปร, ค่าคงที่ หรือ expressions แยกจากกันด้วยเครื่องหมาย comma ใน output-list อาจจะมี iterative DO ได้เช่นเดียวกับที่แสดงไว้ ใน input-list

10.1 List directed I/O

input stream สำหรับ list-directed I/O จะต้องมีคุณสมบัติดังนี้ Data items ใน stream อาจจะเป็น ค่าคงที่เลขคณิต, ค่าคงที่ชนิด character string, หรือ ค่าคงที่ชนิด bit string

data item แต่ละตัว ต้องตามด้วย ตัวคั่นหนึ่งตัว ซึ่งประกอบด้วย เครื่องหมาย

blanks และ comma 1 ตัว ข้างหน้าและข้างหลังเครื่องหมายนี้ อาจจะมี blanks หรือ หรือ an end of file

Embeded tab (ctl-I) เครื่องจะถือว่าเหมือนกับ เครื่องหมาย blanks ข้อมูลชนิด character string ซึ่งมี blanks ภายในเครื่องหมายคำพูด นอกเหนือจากนี้ เครื่องจะถือว่า blanks หรือ comma เป็นเพียงตัวคั่น (seperator)

ฟิลด์ที่ไม่มีข้อมูล (a null field) ใน input stream ทราบได้โดยเครื่องหมาย comma 1 ตัว ซึ่งเป็น first non-blank character ใน input line หรือโดยเครื่องหมาย comma สองตัวติดกัน อาจจะมี blank ตั้งแต่ 1 ตัวขึ้นไปคั่น หรือไม่มีก็ได้ null field หมายถึง ไม่มีข้อมูลที่จะส่ง ไปยัง data item ใน input-list และค่าของข้อมูลเป้าหมาย (target data item) ยังคงเหมือนเดิม เมื่อ options SKIP ปรากฏ ในคำสั่ง GET, linekarks จะถูกนับ แต่ถ้าไม่มี option SKIP, linemarks จะเป็นเพียงตัวคั่นเท่านั้น

10.2 คำสั่ง GET LIST

คำสั่ง GET LIST ใช้อ่านข้อมูล ที่เป็น list-directed I/O โดยมีรูปแบบดังนี้

```
GET [FILE(fname)][SKIP [(nl)]][LIST (input-list)];
```

คำสั่งนี้ จะต้องมี option อย่างน้อยหนึ่งอย่าง และจะเรียงลำดับกันอย่างไรก็ได้ ยกเว้น option LIST ต้องอยู่ หลังสุด ถ้าไม่มี option FILE เครื่องจะถือว่าเป็น FILE (SYSIN) ถ้าใน option SKIP ไม่มี nl เครื่องจะเป็นไป 1 linemark

หลังจากการส่งข้อมูลทั้งหมดไปยัง ชื่อตัวแปร ใน input-list แล้วตำแหน่งคอลัมน์ ใน input stream อยู่ที่ตัวอักขระ หลังข้อมูลตัวสุดท้าย ที่อ่านไปแล้ว

Character string ใน input stream อาจจะอยู่ภายในเครื่องหมายคำพูด หรือไม่อยู่ก็ได้ ถ้าอยู่ภายในเครื่องหมายคำพูด เครื่องหมายคำพูดนั้น จะไม่ถูกส่งไปยัง

ตัวแปร input เช่นเดียวกับ ค่าคงที่ชนิด bit string เครื่องหมายคำพูดเปิด/ปิด และตัวอักษร B ท้ายสุด จะไม่ถูกส่งไปยังตัวแปร input

Input string ถูกจำกัดให้อยู่ใน บรรทัดเดียว และเฉพาะเครื่องหมายคำพูดตัวหน้าสุด ที่จำเป็นสำหรับ string input จาก คอนโซล (console) เมื่อจบด้วย a carriage control

10.3 คำสั่ง PUT LIST

คำสั่ง PUT LIST ใช้ พิมพ์ (write) ข้อมูล ที่เป็น list-directed I/O โดยมีรูปแบบดังนี้

```
PUT [FILE(fname)] [SKIP(nl)] [PAGE [(P)]] [LIST (output-list)];
```

เหมือนกับในคำสั่ง GET LIST คืออย่างน้อยที่สุดต้องมีหนึ่ง option ปรากฏในคำสั่งนี้ และ option LIST ต้องอยู่ท้ายสุด ถ้าไม่มี option FILE เครื่องจะถือว่าเป็น FILE (SYSPRINT) ถ้าใช้ option SKIP แต่ไม่มี nl เครื่องจะ default nl ให้เป็น 1 ถ้า nl = 0 จะไม่มีการพิมพ์ linemark แต่ตำแหน่งคอลัมน์ จะถูก reset ให้เป็น 1 เมื่อใดก็ตาม ถ้ามีการใช้ option SKIP ตำแหน่งคอลัมน์จะถูก reset ให้เป็นหนึ่ง option PAGE จะใช้ได้เฉพาะกับ PRINT files เท่านั้น ถ้าไม่กำหนด P เครื่องจะ default ให้เป็น 1

หมายเหตุ เมื่อใดก็ตามที่มีการพิมพ์ pagemark ทั้งตำแหน่งคอลัมน์ และเลขประจำบรรทัด จะถูก set ให้เป็น 1

data items ใน output-list จะถูกเปลี่ยนรูป ให้เป็นรูปแบบของ การแทน character string และพิมพ์ ใน STREAM file เราใช้เครื่องหมาย blanks คั่น ข้อมูลบน output file อย่างไรก็ตาม ถ้า data item ยาวกว่า จำนวน ของตัวอักขระทางซ้ายมือของ output line, item นั้นจะถูกพิมพ์ ที่ตำแหน่งเริ่มต้น ของบรรทัดถัดไป ถ้าความยาวของ การแทน character string ของ data item มากกว่าขนาดของบรรทัด data item จะถูกพิมพ์ โดยตัวมันเอง บน บรรทัดเดียว ซึ่ง ขยาย ขนาดของบรรทัด ถ้าขนาดหน้า

กระดาษมากเกินไประหว่างการส่ง output จะเกิดเงื่อนไข ENDPAGE ขึ้น

ปกติ character string จะถูกพิมพ์ ภายในเครื่องหมายคำพูด ด้วย embeded quote symbol แต่ละตัว ซึ่งพิมพ์เป็น เครื่องหมายคำพูด 1 คู่ ถ้าแฟ้มข้อมูลนั้น มี attribute PRINT เครื่องหมายคำพูดที่เกินมาจะถูกตัดทิ้ง ข้อมูลชนิด bit string จะพิมพ์ภายในเครื่องหมายคำพูดเสมอ ตามด้วยตัวอักษร B

10.4 EDIT-directed I/O

input-list และ output-list สำหรับ edit-directed I/O จะเหมือนกับ ใน list-directed I/O อย่างไรก็ตาม ลักษณะการอ่านข้อมูล หรือการพิมพ์ข้อมูล กำหนด โดย รายชื่อของ format items ใน format-list ของคำสั่ง GET EDIT และคำสั่ง PUT EDIT

10.5 FORMAT-list

format-list เป็น รายชื่อของ format items ซึ่งคั่นด้วยเครื่องหมาย comma บรรยายลักษณะของ data items ซึ่งจะถูกอ่าน (หมายถึง data format items) และกำหนด ตำแหน่งของ data items ใน stream(หมายถึง control format items) หรือการอ้างถึง format-list อีกชุดหนึ่ง (หมายถึง remote format item) โดยมีรูปแบบดังนี้

```
[n] f-item . . . [,[n] f-item]
```

เมื่อ n เป็นค่าคงที่ชนิด literal อยู่ในช่วง 1 ถึง 254 หมายถึง จำนวนซ้ำ (repetition factor) ของ f-item ตัวที่ตามหลัง ถ้าไม่มี n เครื่อง จะถือว่า จำนวนซ้ำ ของ f-item ตัวที่ตามหลังมีค่าเท่ากับ 1

f-item อาจจะเป็น data format item หรือ control format item ก็ได้ ในการบอกจำนวนซ้ำ ของ format items จำนวนหนึ่ง, f-item ตัวนั้น

สามารถทำให้เป็น group ได้ดังนี้

(format-list)

ใน PL/I-80 f-item แต่ละตัว อาจจะเป็น remote format item ก็ได้ อย่างไรก็ตาม remote format item แต่ละตัว ต้องเป็นเพียง format ใน list เท่านั้น และจะมี repetition factor นำหน้าไม่ได้

10.6 Data format items

data format item ใช้สำหรับอ่าน หรือพิมพ์ ฟิลด์ที่เป็น ตัวเลข หรือฟิลด์ที่เป็นตัวอักขระ จาก external STREAM data set หรือไปยัง external STREAM data set ใน PL/I-80 มี data format items ดังนี้

F(w[,d]) ใช้กับ fixed point arithmetic data

เมื่อ w หมายถึงความกว้าง (จำนวนตัวอักขระในฟิลด์นั้น) และ d คือจำนวนอักขระทางขวามือของจุดทศนิยม

สำหรับ input จำนวนตัวอักขระที่กำหนดโดย w จะถูกอ่าน ถ้า character string นั้นมีจุดทศนิยม จุดทศนิยมจะเป็นตัวกำหนด scale ถ้าไม่มีจุดทศนิยม d จะเป็นตัวกำหนด scale เครื่องหมาย blanks หน้าสุด หรือ ท้ายสุด เครื่องไม่สนใจ ถ้าฟิลด์นี้ มีแต่ character blanks เท่านั้น ค่าที่อ่านคือศูนย์

สำหรับ output d จะเป็นตัวกำหนด scale ของค่า output ถ้าไม่มี d, scale จะมีค่าเป็นศูนย์ ค่าของ output จะถูกปัดเศษ (rounded) ถ้าตัวแปรนั้น มี precision ไม่เท่ากับ 15 (ค่าสูงสุดของ precision) เลขศูนย์ข้างหน้าตัวเลข จะถูกตัดทิ้ง ยกเว้นเลขศูนย์ 1 ตัว หน้าจุดทศนิยม

E(w[,d]) ใช้กับ output เพื่อแทน ข้อมูลเลขคณิต ใน รูปแบบสัญลักษณ์ทางวิทยาศาสตร์

สำหรับ input เพื่อเปลี่ยนรูป decimal characters ให้เป็นค่า float binary w หมายถึงความกว้างของฟิลด์ ขณะที่ d เป็นตัวกำหนด จำนวนตัวเลขทางขวามือของจุดทศนิยม

สำหรับ output w ต้องมากกว่า d เท่ากับ 7 เนื่องจาก output ฟิลด์จะปรากฏดังนี้

```
+n.ddddE+ee
```

เมื่อ + แทนตำแหน่งของเครื่องหมาย, n เป็นเลขหน้าสุด dddd แทน fractional part ของ ความยาว d และ E+ee แทน exponent ฟิลด์

A[(w)] ใช้สำหรับอ่านหรือพิมพ์ ตัวอักขระจำนวน w ตัวของ ข้อมูลชนิด character string

สำหรับ input w ต้องใช้กับ PL/I ชุดสมบูรณ์ได้ด้วย อย่างไรก็ตาม ใน PL/I-80 สำหรับ input อาจจจะไม่ต้องมี w ก็ได้ ในกรณีนี้จะอ่านจนถึงส่วนที่เหลือของบรรทัดปัจจุบัน แต่ไม่นับ carriage return line feed

สำหรับ output ถ้าไม่มี w เครื่องจะถือว่า w คือความยาวของ output string ถ้า w มีค่ามากกว่า ความยาวของ string output เครื่องจะเติม blanks ให้ทางขวามือ ถ้า w มีค่าน้อยกว่า ความยาวของ output string เครื่องจะตัดตำแหน่งทางขวามือของ string

B[b][(w)] หมายถึงการแทนข้อมูลชนิด bit string

สำหรับ input จะต้องมี w อยู่ด้วย และใน input stream นั้นจะมีเฉพาะเลข 0 หรือ เลข 1 เท่านั้น ถ้าเป็นอย่างอื่น (otherwise) จะเกิดเงื่อนไข ERROR ขึ้น จำนวนบิทที่จะถูกใช้สำหรับตัวเลขแต่ละตัว ถูกกำหนดโดย b

สำหรับ output ตัวแปรจะถูกเปลี่ยนรูป ให้เป็นชนิด bit string จากนั้นจึงเปลี่ยนรูปเป็นการแทนที่ของ character string ของมัน ถ้าไม่กำหนด w, character string ผลลัพธ์ คือ output ถ้ากำหนด w และค่าของ w มากกว่า character string เครื่องจะใส่ blanks ให้ทางขวามือ แต่ถ้า character string ผลลัพธ์ ยาวกว่า w จะเกิดเงื่อนไข ERROR ขึ้น

10.7 Control format items

control format items ใช้สำหรับการใส่ line, page และ space ใน PL/I-80 มี control format items ดังนี้

COLUMN(nc) หรือ COL(nc) ย้ายตัวชี้รูปแบบ (format pointer) ไปยังคอลัมน์ nc ใน input data stream หรือ output data stream

สำหรับ input ตัวอักขระต่าง ๆ ซึ่งถูกข้ามไปจนถึงตำแหน่งคอลัมน์ nc เครื่องจะไม่สนใจ ถ้าตำแหน่งคอลัมน์ปัจจุบัน น้อยกว่า nc ตัวชี้รูปแบบจะย้ายไปยังตำแหน่งคอลัมน์ nc แต่ถ้าตำแหน่งคอลัมน์ปัจจุบันมากกว่า nc ตัวชี้จะย้ายไปยังบรรทัดถัดไป จากนั้นจึงย้ายไปยังตำแหน่งคอลัมน์ nc อันใหม่ ถ้า nc ยาวกว่าตำแหน่งขวามือสุดของบรรทัด ตัวชี้รูปแบบจะย้ายไปยังคอลัมน์แรกของบรรทัดใหม่

สำหรับ input การย้ายของตัวชี้รูปแบบ ตัด input characters ทิ้ง แต่สำหรับ output เครื่องจะใส่ blanks ไว้ใน stream ขณะที่มีการย้ายตัวชี้ไปข้างหน้า

สำหรับ output เครื่องจะพิมพ์ blanks ใน process ของการย้ายไปยังตำแหน่ง nc เช่นเดียวกัน ถ้าตำแหน่งปัจจุบัน มากกว่า ตัวเลข nc โปรแกรมจะให้ output เป็น linemark แล้วพิมพ์ blanks จนกระทั่งถึงคอลัมน์ nc ของบรรทัดใหม่ ถ้า nc ยาวกว่าขนาดของบรรทัด เครื่องจะพิมพ์ linemark แล้วตำแหน่งคอลัมน์ จะถูก set ให้เป็น 1

X(sp) ย้ายตัวชี้รูปแบบไปข้างหน้า sp ตำแหน่ง ใน input data streams หรือ output data stream

สำหรับ input sp คือจำนวนตัวอักขระที่จะให้เว้นไปข้างหน้า ถ้าเจอ linemark เครื่องจะไม่สนใจ การปฏิบัติการ จะทำต่อบนบรรทัดถัดไป

สำหรับ output sp คือจำนวน blanks ที่จะให้พิมพ์ ถ้าจบบรรทัดก่อนการปฏิบัติการใส่ blank จะทำต่อบนบรรทัดถัดไป

SKIP[(nl)] ใช้สำหรับกำหนด จำนวนของ linemarks nl ที่จะทำให้ข้ามไป หรือให้พิมพ์ ถ้าไม่กำหนด nl เครื่องจะถือว่าเท่ากับ 1 และตำแหน่งคอลัมน์ จะถูก set ให้เป็น 1

สำหรับ input nl คือจำนวน linemarks ที่จะให้ข้ามไปก่อน การย้ายไปยัง format item ถัดไป โปรดสังเกตว่า ถ้า SKIP(1) ถูก execute ขณะที่ format item แรก ตามหลัง explicit หรือ implicit การปฏิบัติการ OPEN ทันที บรรทัดแรกจะถูกตัดทิ้ง (discarded) นอกจากนี้แล้ว SKIP(0) เป็น undefined สำหรับ input stream

สำหรับ output nl คือจำนวน linemarks ที่จะให้พิมพ์ ถ้าใน process ของการพิมพ์ linemarks ขนาดหน้ากระดาษมากเกินไปสำหรับ PRINT file จะเกิดเงื่อนไข ENDPAGE ขึ้น ซึ่งส่งมาจาก ON-unit ดังนั้นการปฏิบัติการ SKIP จึงต้องหยุด (aborted)

LINE(ln) ใช้เฉพาะกับ PRINT file เท่านั้น และกำหนดเลขประจำบรรทัดของ data item ถัดไปที่จะให้พิมพ์ ค่าคงที่ ln ต้องมากกว่าศูนย์

ถ้าเลขประจำบรรทัดปัจจุบัน เท่ากับ ln จะไม่มีปฏิบัติการอะไรเกิดขึ้น ถ้าเลขประจำบรรทัดปัจจุบัน น้อยกว่า ln output คือ linemarks จนกระทั่งเลขประจำบรรทัดเท่ากับ ln ถ้า linemarks ที่กำหนดนั้น มากกว่าขนาดหน้ากระดาษปัจจุบัน จะเกิดเงื่อนไข ENDPAGE ขึ้น

PAGE ใช้เฉพาะกับ PRINT file เท่านั้น option PAGE ทำให้มีการพิมพ์ page-mark เลขประจำหน้าจะเพิ่มขึ้นทีละ 1 เลขประจำบรรทัดและตำแหน่งคอลัมน์ จะถูก set ให้เป็น 1

หมายเหตุ control format items จะถูก execute เมื่อมันถูกพบใน format-list สำหรับ control format items ใด ๆ ก็ตามถ้าเกิดหลังจาก input-list หรือ output-list หมดไปแล้ว จะไม่เกิดผลแต่อย่างใด (no effect)

10.8 Remote Format Items

Remote format item จะใช้ format-list ของคำสั่ง FORMAT แทนที่ในตำแหน่งของ remote format item, โดยมีรูปแบบดังนี้

```
R (format-label)
```

เมื่อ format-label เป็นค่าคงที่ label อยู่ข้างหน้า คำสั่ง FORMAT ซึ่งอยู่ในขอบเขตของ remote format item ตามที่กล่าวข้างต้น PL/I-80 มีข้อจำกัดว่าเฉพาะ remote format item สามารถปรากฏเฉพาะโดยตัวมันเอง ใน format-list และต้องไม่มี repetition factor นำหน้า

ตัวอย่าง

```
PUT SKIP EDIT (A,B,C)(R(ELSEWHERE));
```

10.9 คำสั่ง FORMAT

คำสั่งนี้ ใช้ นิยาม remote format item โดยมีรูปแบบดังนี้

```
format-label:   FORMAT   (format-list);
```

เมื่อ format-label เป็นค่าคงที่ label ที่สมนัยกับ FORMAT

format-list เป็นรายชื่อของ format items ตามที่กล่าวมาแล้วในหัวข้อก่อนหน้านี้

ตัวอย่าง

```
L1: FORMAT (A(5),F(6,2),SKIP(3),A(2));
```

และจะถูกอ้างถึงเป็น remote format ในคำสั่งข้างล่างนี้

```
GET EDIT (A,B,C) (R(L1));
```

10.10 คำสั่ง GET EDIT

คำสั่งนี้ อ่านข้อมูล โดยการใช้ format-list โดยมีรูปแบบดังนี้

```
GET [FILE (fname)] SKIP [(n1)] [EDIT (input-list)(format-list)];
```

เหมือนกับในคำสั่ง GET LIST คือ data items ถูกอ่านเข้าไปเก็บในตัวแปร ที่กำหนดใน input-list จนกระทั่ง input-list หมดหรือ จนถึง end of file คำสั่ง GET EDIT อย่างไรก็ตาม จะจับคู่ input-list item กับ sequential format-list item ถัดไป การใช้ control format item จะเกิดผลใน process นั้น ถ้า input-list หมดก่อน format-list, format items ที่เหลือ เครื่องจะไม่สนใจ แต่ถ้า format-list หมดก่อน input-list, format-list จะถูก process ใหม่อีกครั้งหนึ่งจากตำแหน่งเริ่มต้น

10.11 คำสั่ง PUT EDIT

คำสั่งนี้พิมพ์ output data items ตามที่กำหนดใน format-list โดยมีรูปแบบดังนี้

```
PUT [FILE (fname)][SKIP [(n)]][PAGE [(p)]]
        [EDIT      (output-list)(format-list)];
```

เหมือนกับในคำสั่ง GET EDIT จะมีการจับคู่ output expression จาก output-list กับ format items จาก format-list ใน process นี้ อาจจะมี control format items อยู่ด้วย ถ้าคำสั่ง PUT เสร็จสิ้นแล้ว แต่ยังมี format items ที่ยังไม่ได้ถูกใช้ เครื่องจะไม่สนใจ นอกจากนี้แล้ว ถ้า format-list หมดก่อนระหว่างการประมวล-

ผล เครื่องจะใช้ format-list นั้นซ้ำอีก โดยเริ่มตั้งแต่ต้น

10.12 Line-Directed I/O

ทั้งสองรูปแบบของคำสั่ง READ และคำสั่ง WRITE นำมาใช้ได้ใน PL/I-80 สำหรับการประมวลผล variable length ASCII records ในแฟ้มข้อมูลชนิด STREAM รูปแบบเหล่านี้ จะนำไปใช้ทั่วไป ใน PL/I ชุดอื่น ๆ ไม่ได้ และควรจะหลีกเลี่ยงถ้า upward compatibility มีความสำคัญ, ทั้งสองรูปแบบนี้ มีชื่อว่า READ Varying และ WRITE Varying

10.13 คำสั่ง READ Varying

คำสั่งนี้ ใช้สำหรับอ่าน variable length STREAM INPUT files โดยมีรูปแบบดังนี้

```
READ [FILE (fname)] INTO (v);
```

เมื่อ v เป็นตัวแปรชนิด CHAR VARYING string

ถ้าไม่มี option FILE เครื่องจะถือว่าเป็น FILE(SYSIN) สิ่งสำคัญที่จะต้องจำคือ คำสั่ง READ ที่จะกล่าวถึง ในหัวข้อต่อไปนี้ แตกต่างจาก รูปแบบของ READ Varying เฉพาะที่ ตัวแปรเป้าหมายต้องมี VARYING attribute เท่านั้น

ข้อมูลถูกอ่านจากแฟ้มข้อมูล จนกระทั่งถึง ความยาวสูงสุดของ v หรือ line-feed character ถูกอ่าน ความยาวของ v ถูกกำหนดให้เป็น จำนวนตัวอักขระที่อ่าน รวมทั้ง line-feed character

หมายเหตุ เมื่อคำสั่ง READ Varying ทำให้เกิดการ default OPEN, attribute ของแฟ้มข้อมูลผลลัพธ์ จะมี STREAM INPUT

ตัวอย่าง การ declare

```
1 BUFFER,
  2 BUFFCH CHAR(254) VAR;
```

คำสั่ง READ FILE(F) INTO (BUFFER);

หมายถึง การเคลื่อนย้ายข้อมูลชนิด RECORD oriented เนื่องจากเป้าหมายเป็นโครงสร้าง ไม่ใช่ ตัวแปรชนิด CHAR VARYING

คำสั่ง READ FILE(F) INTO (BUFFCH);

นั่นคือ มีความหมายเป็น การเคลื่อนย้ายข้อมูลชนิด ASCII STREAM INPUT เนื่องจากเป้าหมายเป็น CHAR VARYING

10.14 คำสั่ง WRITE VARYING

คำสั่งนี้ ใช้สำหรับพิมพ์ (write) variable-length ASCII STREAM data โดยมีรูปแบบดังนี้

```
WRITE [FILE (fname)] FROM (v);
```

เมื่อ v เป็นตัวแปรชนิด CHAR VARYING string

ห้ามใส่ control characters ใน output string ถ้าจำเป็นต้องมี control characters มันจะรวมอยู่ในส่วนหนึ่งของ string โปรดระลึกไว้เสมอว่า PL/I-80 อนุญาตให้ embeded control character เป็น ส่วนหนึ่งของค่าคงที่ string สัญลักษณ์ที่ใช้อยู่หน้า "^" ภายใน string

คำสั่ง WRITE VARYING ใช้ไม่ได้ทั่วไป ใน PL/I และต้องหลีกเลี่ยงถ้าจำเป็นต้องมี upward compatibility เช่นเดียวกับในกรณีของ READ Varying, WRITE และ WRITE VARYING แตกต่างกัน ตรงที่ว่า ตัวแปร source มี attribute VARYING นอกจากนี้ คำสั่งนี้ จะให้ default OPEN รวมทั้ง attribute STREAM OUTPUT

ตัวอย่าง เมื่อกำหนดการ declare เหมือนในหัวข้อก่อนหน้านี้

คำสั่ง WRITE FILE(F) FROM (BUFFER);

เป็นการเคลื่อนย้ายข้อมูลชนิด RECORD oriented ในขณะที่

คำสั่ง WRITE FILE(F) FROM (BUFFCH);

เป็นคำสั่ง WRITE VARYING ปฏิบัติการบนแฟ้มข้อมูลชนิด ASCII STREAM OUTPUT เนื่องจาก ตัวแปร source มี attribute VARYING