



บทที่ 3

การประยุกต์คำสั่งปฐมแบบซีเอสพีเข้ากับภาษาซี

จากการศึกษาคำสั่งปฐมแบบต่าง ๆ ของซีเอสพี และ ลักษณะของคำสั่งในภาษาซี ทำให้สามารถสรุปได้ว่า การที่จะพัฒนาเครื่องมือทางซอฟต์แวร์ที่ทำหน้าที่เหมือนคำสั่งปฐมแบบซีเอสพีได้ แบ่งออกเป็น 2 ลักษณะ คือ

1. พัฒนาฟังก์ชันที่ให้โปรแกรมประยุกต์เรียกใช้
2. กำหนดรูปแบบการเขียนกลุ่มของคำสั่ง (Programming Convention) ขึ้น

ฟังก์ชันที่ต้องพัฒนาขึ้นเพื่อให้โปรแกรมประยุกต์เรียกใช้

ฟังก์ชันที่ต้องพัฒนาขึ้นมี 3 ฟังก์ชันหลัก คือ

1. ฟังก์ชันซีเอสพีพาร์ (cspPar function)
ทำหน้าที่เหมือนคำสั่งพาราเลสของซีเอสพี
2. ฟังก์ชันซีเอสพีอิน (cspIn function)
ทำหน้าที่เหมือนคำสั่งอินพุทของซีเอสพี
3. ฟังก์ชันซีเอสพีเอาท์ (cspOut function)
ทำหน้าที่เหมือนคำสั่งเอาท์พุทของซีเอสพี

ฟังก์ชันซีเอสพีพาร์

ทำหน้าที่เช่นเดียวกับคำสั่งพาราเลลของซีเอสพี นั่นคือ กำหนดการประมวลผลแบบจวนกันของโปรเซสต่าง ๆ ที่ระบุ

รูปแบบ

```
int cspPar(param)
char *param[];
```

ฟังก์ชันซีเอสพีพาร์จะทำการตรวจสอบความถูกต้องทางวากยสัมพันธ์ (syntax) (วิศวกรรมสถานแห่งประเทศไทย, 2527-2530) ของชื่อโปรเซสแต่ละโปรเซส (ตัวอย่างเช่น มีวงเล็บหรือไม่ ถ้ามีวงเล็บเปิดแล้วมีวงเล็บปิดหรือไม่ หรือ ถ้ามีการกำหนด subscript เป็นช่วงได้กำหนดถูกต้องหรือไม่) และ ตรวจสอบว่าชื่อโปรเซสนั้น ๆ ได้ถูกระบุไว้ในโครงสร้างชื่อ funcMap (ที่ใช้กำหนดว่าจะประมวลผลฟังก์ชันใด) หรือไม่ ถ้าถูกต้อง จะทำการสร้างโปรเซสนั้น ๆ ขึ้นมาประมวลผล และ เก็บชื่อโปรเซสพร้อมทั้งหมายเลขโปรเซสไว้ในโครงสร้าง procMap จากนั้น จะวนกลับไปทำเช่นเดิมกับชื่อโปรเซสถัดไปจนกว่าจะครบทุกชื่อ แต่ถ้าไม่ถูกต้องจะทำการหยุดการประมวลผลทุกโปรเซสที่ถูกสร้างขึ้น

ในการสร้างโปรเซสขึ้นมาประมวลผลนั้น จะมีตัวแปรชื่อ NPROC ที่กำหนดจำนวนโปรเซสสูงสุดที่จะสร้างขึ้นเพื่อประมวลผลพร้อมกัน ในกรณีที่จำนวนชื่อโปรเซสที่ระบุมีมากกว่าค่าที่กำหนดไว้ ฟังก์ชันนี้จะหยุดการประมวลผลทุกโปรเซสที่ถูกสร้างขึ้นเช่นกัน

ฟังก์ชันซีเอสพีพาร์จะหยุดการประมวลผลเมื่อทุกโปรเซสที่ถูกสร้างขึ้นหยุดการประมวล

ผลลง

ค่าที่ฟังก์ชันซีเอสพีพาร์จะส่งกลับ มีดังนี้ คือ

- ก) SUCCESS แสดงว่าการประมวลผลฟังก์ชันนี้เสร็จสมบูรณ์
- ข) ERROR มีข้อผิดพลาดบางประการในการประมวลผล เช่น ชื่อโปรเซสไม่ถูกวากยสัมพันธ์ หรือ ชื่อโปรเซสบางโปรเซสมิได้ถูกระบุในโครงสร้าง funcMap หรือ จำนวนโปรเซสมีมากกว่าจำนวนที่กำหนดในตัวแปร NPROC

ฟังก์ชันซีเอสพีอิน

ทำหน้าที่เช่นเดียวกับคำสั่งอินพุทของซีเอสพี นั่นคือ รอรับค่าข้อมูลจากโปรเซส
ต้นทางที่ระบุ มาเก็บไว้ในตัวแปรที่ระบุ

รูปแบบ

```
int cspIn(s_name, int_value, flag)
char s_name[];
int *int_value, flag;
```

ฟังก์ชันซีเอสพีอินจะทำการตรวจสอบความถูกต้องทางวากยสัมพันธ์ของชื่อโปรเซสต้น
ทาง (s_name) ที่ระบุ (ตัวอย่าง เช่น มีวงเล็บหรือไม่ ถ้ามีวงเล็บเปิดแล้วมีวงเล็บปิด
หรือไม่ หรือ ถ้ามีการกำหนด subscript เป็นนิพจน์ ได้กำหนดถูกต้องหรือไม่) และ
ตรวจสอบว่าชื่อโปรเซสนั้น ๆ ได้ถูกระบุไว้ในโครงสร้างชื่อ procMap (ที่เก็บชื่อโปรเซสที่ถูก
สร้างขึ้น) หรือไม่ ถ้าถูกต้องจะทำการรับค่าจากโปรเซสต้นทางที่ระบุมาเก็บในตัวแปร
int_value หลังจากนั้น จะส่งข้อความแจ้งให้โปรเซสต้นทางทราบ

ในการรับค่าจากโปรเซสต้นทาง ถ้าโปรเซสต้นทางยังมีได้ประมวลผลฟังก์ชันซีเอสพี
เอาท์ที่สอดคล้องกัน ฟังก์ชันซีเอสพีอินอาจรอหรือไม่รับค่า ทั้งนี้ขึ้นอยู่กับค่าของตัวแปร
flag ถ้ามีค่าเป็น 0 จะรอรับค่า แต่ถ้ามีค่าเป็น IPC_NOWAIT จะไม่รอ

เทคนิคที่ใช้ในการรับค่าจากโปรเซสอื่น มีอยู่หลายวิธี (Bach, M.J., 1986) เช่น
ไพล์ไลน์ (pipeline) การส่งผ่านข่าวสาร (message passing) แต่วิธีที่ใช้ในการ
พัฒนาเครื่องมือทางซอฟต์แวร์นี้ คือ การส่งผ่านข่าวสาร เช่นเดียวกับวิธีที่ Roper, T.J.
& Barter, C.J. (1978) และ Gentleman, W. Morven (1981) ใช้

ค่าที่ฟังก์ชันซีเอสพีอินจะส่งกลับ มีดังนี้ คือ

- ก) COMM_OK แสดงว่าการประมวลผลฟังก์ชันนี้เสร็จสมบูรณ์
- ข) COMM_NOK มีข้อผิดพลาดบางประการอันเนื่องมาจาก system call ที่ใช้
ในการรับค่า
- ค) COMM_DIE โปรเซสต้นทางที่ระบุมิได้ประมวลผลอยู่ในขณะนั้น
- ง) COMM_ERR มีข้อผิดพลาดบางประการในการประมวลผล เช่น ชื่อโปรเซส
ไม่ถูกวากยสัมพันธ์ หรือ ชื่อโปรเซสต้นทางมิได้ถูกระบุในโครงสร้าง procMap

ฟังก์ชันซีเอสพีเอช

ทำหน้าที่เช่นเดียวกับคำสั่งเอชพีชของซีเอสพี นั่นคือ ส่งค่าข้อมูลในตัวแปรที่ระบุไปยังโปรเซสปลายทางที่ระบุ

รูปแบบ

```
int cspOut(d_name, int_value)
char d_name[];
int *int_value;
```

ฟังก์ชันซีเอสพีเอชจะทำการตรวจสอบความถูกต้องทางวากยสัมพันธ์ของชื่อโปรเซสปลายทาง (d_name) ที่ระบุ (ตัวอย่าง เช่น มีวงเล็บหรือไม่ ถ้ามีวงเล็บเปิดแล้วมีวงเล็บปิดหรือไม่ หรือ ถ้ามีการกำหนด subscript เป็นนิพจน์ ได้กำหนดถูกต้องหรือไม่) และตรวจสอบว่าชื่อโปรเซสนั้น ๆ ได้ถูกระบุไว้ในโครงสร้างชื่อ procMap (ที่เก็บชื่อโปรเซสที่ถูกสร้างขึ้น) หรือไม่ ถ้าถูกต้องจะทำการส่งค่าของตัวแปร int_value ไปยังโปรเซสปลายทางที่ระบุ และ จะรอการแจ้งกลับจากโปรเซสปลายทาง

ในการส่งค่าไปยังโปรเซสปลายทาง ฟังก์ชันซีเอสพีเอชจะรอจนกว่าโปรเซสปลายทางจะประมวลผลฟังก์ชันซีเอสพีอินที่สอดคล้องกัน

เช่นเดียวกับฟังก์ชันซีเอสพีอิน วิธีที่ใช้ในการส่งค่าข้อมูล คือ การส่งผ่านข่าวสารค่าที่ฟังก์ชันซีเอสพีเอชจะส่งกลับ มีดังนี้ คือ

- ก) COMM_OK แสดงว่าการประมวลผลฟังก์ชันนี้เสร็จสมบูรณ์
- ข) COMM_NOK มีข้อผิดพลาดบางประการอันเนื่องมาจาก system call ที่ใช้ในการส่งค่า
- ค) COMM_DIE โปรเซสปลายทางที่ระบุมิได้ประมวลผลอยู่ในขณะนั้น
- ง) COMM_ERR มีข้อผิดพลาดบางประการในการประมวลผล เช่น ชื่อโปรเซสไม่ถูกวากยสัมพันธ์ หรือ ชื่อโปรเซสปลายทางมิได้ถูกระบุในโครงสร้าง procMap

รูปแบบการเขียนกลุ่มของคำสั่ง

เนื่องจากภาษาซีไม่คล่องตัวพอที่จะทำคำสั่งการ์ดเด็ด คำสั่งอัลเทอเนทีฟ และ คำสั่งรีเพททีฟ ในรูปของไลบรารีฟังก์ชันได้โดยตรง จึงได้มีการกำหนดรูปแบบการเขียนกลุ่มของคำสั่งขึ้น 2 รูปแบบ ดังนี้

1. รูปแบบเงื่อนไข
2. รูปแบบการทำซ้ำ

รูปแบบเงื่อนไข

ทำหน้าที่เหมือนคำสั่งอัลเทอเนทีฟ และ คำสั่งการ์ดเด็ดของซีเอสพี

รูปแบบ

```

if condition
    command
else
    if condition
        command
    else
        ...

```

แต่ละเงื่อนไข (condition) เทียบได้กับการ์ดของคำสั่งการ์ดเด็ดของซีเอสพี ส่วนแต่ละคำสั่งงาน (command) ซึ่งอาจเป็นคำสั่งเดียว หรือ หลายคำสั่งอยู่ในเครื่องหมาย { } (ตามกฎเกณฑ์ของภาษาซี) เทียบได้กับคำสั่งที่ต้องการประมวลผลซึ่งอยู่หลัง เครื่องหมาย "->" ของคำสั่งการ์ดเด็ด และ แต่ละ if เทียบได้กับคำสั่งการ์ดเด็ดของคำสั่งอัลเทอเนทีฟของซีเอสพี

ในกรณีที่มีเงื่อนไขที่เป็นจริงในขณะเดียวกันมากกว่า 1 เงื่อนไข เครื่องมือทางซอฟต์แวร์นี้จะเลือกทำเงื่อนไขที่ถูกระบุไว้ก่อน ดังนั้น จึงไม่มีหลักการของ nondeterminism

ตัวอย่างการเขียนกลุ่มคำสั่งในรูปแบบเงื่อนไข

ซีเอสพี (จากบรรทัดที่ 8 และ 9 ของโปรแกรมตัวอย่างข้างต้น) :

```
[m = mp -> skip   []
m < mp -> SIEVE(i+1)!m ]
```

ภาษาซี :

```
if (m == mp)
    ;
else
    if (m < mp)
        cspOut("SIEVE(i+1)", &m);
    else
        ;
```

รูปแบบการทำซ้ำ

ทำหน้าที่เหมือนคำสั่งรีเพททิฟของซีเอสพี

รูปแบบ

```
while (true)
    if condition
        command
    else
        break
```


if เทียบได้กับคำสั่งอัลเทอเนทีฟของซีเอสพี ซึ่งอยู่ภายในคำสั่งรีเพททีฟ การทำซ้ำนี้จะกระทำจนกว่าเงื่อนไข (condition) ไม่เป็นจริง จึงจะจบการทำงานคำสั่งนี้

ตัวอย่างการเขียนกลุ่มคำสั่งในรูปแบบการทำซ้ำ

ซีเอสพี (จากบรรทัดที่ 12 ของโปรแกรมตัวอย่างข้างต้น) :

```
*[n < 10000 -> SIEVE(1)!n; n:=n+2]
```

ภาษาซี :

```
while (true)
    if (n < 10000) {
        cspOut("SIEVE(1)", &n);
        n += 2;
    }
    else
        break;
```