หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2
คำสั่งปฐมแบบซีเอสพี

ซีเอสพีเป็นเทคนิคที่ Hoare, C.A.R (1978) แนะนำขึ้นเพื่อให้การประมวลผลเป็นไปได้เร็วขึ้น โดยแบ่งงานที่ต้องการประมวลผลออกเป็นโปรเซสหลายโปรเซสที่เริ่มประมวลผลพร้อมกัน แต่สามารถแยกกันประมวลผลได้อย่างเป็นอิสระ และ แต่ละโปรเซสมิได้ใช้หน่วยความจำร่วมกัน ในการทำงาน แต่ละโปรเซสสามารถติดต่อกันได้โดยการส่งผ่านค่าข้อมูลซึ่งกันและกัน ภายในแต่ละโปรเซสยังคงไว้ซึ่งโครงสร้างของการเขียนโปรแกรม 3 โครงสร้าง คือ โครงสร้างการประมวลผลแบบเรียงลำดับ โครงสร้างการเลือก และ โครงสร้างการทำซ้ำ

ตัวอย่างโปรแกรมซีเอสพี

ตัวอย่างโปรแกรมต่อไปนี้ ได้จากบทความของ Hoare, C.A.R (1978)

```
1   [SIEVE(i:1..100)::
2         p,mp:integer;
3         SIEVE(i-1)?p;
4         print!p;
5         mp:=p;  comment mp is a multiple of p;
6         *[m:integer;  SIEVE(i-1)?m ->
7               *[m > mp  ->  mp :- mp + p];
8                [m = mp  ->  skip   []
9                 m < mp  -> SIEVE(i+1)!m ]
10        ]
11  ||  SIEVE(0)::print!2; n:integer; n:=3;
12        *[n < 10000  ->  SIEVE(1)!n;  n:=n+2]
13  ||  SIEVE(101)::*[n:integer;  SIEVE(100)?n  ->  print!n]
```

```
14   ||  print::*[(i:0..101)n:integer;  SIEVE(i)?n  ->  ... ]
15   ]
```

โปรแกรมข้างต้นทำการหาค่า prime number 101 ตัวที่อยู่ระหว่าง 1 และ 10000 โดย SIEVE(0) ถึง SIEVE(101) จะทำการหาค่า เพื่อส่งให้ print พิมพ์ ในการทำงาน SIEVE(0) จะส่งค่า 2 ซึ่งเป็น prime number ตัวแรกให้ print และกรองค่าเลขคู่ (ซึ่งหารด้วย 2 ลงตัว) ออก ส่วนค่าที่เหลือส่งให้ SIEVE(1) ซึ่งจะส่งค่าตัวแรกที่ได้รับ คือ 3 (prime number ตัวที่ 2) ให้ print และ กรองค่าที่หารด้วย 3 ลงตัวออก และ ส่งค่าที่เหลือให้ SIEVE(2) ซึ่งจะมีการทำงานเช่นเดียวกัน สำหรับ SIEVE(101) ซึ่งได้รับค่าที่หารด้วย prime number 101 ตัวแรกไม่ลงตัว จะส่งทุกค่าที่ได้รับจาก SIEVE(100) ให้ print ทั้งหมด

<u>คำอธิบายโปรแกรมตัวอย่าง</u>

บรรทัดที่ 1 ระบุโปรเซสชื่อ SIEVE ซึ่งมี 100 โปรเซส ตั้งแต่ 1 ถึง 100 ชุดของโปรเซสนี้ประกอบด้วยชุดของคำสั่งชุดเดียวกัน ซึ่งอยู่ตามหลังเครื่องหมาย :: โดยเริ่มตั้งแต่บรรทัดที่ 2 ถึง 10

บรรทัดที่ 2 กำหนดตัวแปร 2 ตัว คือ p และ mp เป็นชนิดเลขจำนวนเต็ม

บรรทัดที่ 3 คำสั่งอินพุท (Input Command) ของซีเอสพี จะรอรับค่าข้อมูลจากโปรเซสชื่อ SIEVE ที่มี subscript ก่อนหน้าโปรเซสนี้ และ เก็บค่าข้อมูลที่ได้ไว้ในตัวแปรชื่อ p

บรรทัดที่ 4 คำสั่งเอาท์พุท (Output Command) ของซีเอสพี จะส่งค่าข้อมูลในตัวแปรชื่อ p ให้โปรเซสชื่อ print โดยจะรอจนกว่าโปรเซส print พร้อมจะรับข้อมูลจากโปรเซสนี้

บรรทัดที่ 5 ให้ mp มีค่าเท่ากับค่าของ p

บรรทัดที่ 6-10 คำสั่งรีเพททีทีฟ (Repetitive Command) ของซีเอสพี ซึ่งระบุโดยเครื่องหมาย * หน้าคำสั่งที่ครอบด้วย "[" และ "]" กำหนดตัวแปรชื่อ m เป็นชนิดเลขจำนวนเต็ม และ ทดสอบว่าโปรเซส SIEVE ที่มี subscript ก่อนหน้าโปรเซสนี้ส่งค่าข้อมูลมาให้หรือไม่ ถ้าส่งมาจะประมวลผลคำสั่งในบรรทัดที่ 7 ถึง 9 มิฉะนั้น จะหยุดการประมวลผลคำสั่งรีเพททีทีฟนี้ และ ไปประมวลผลคำสั่งในบรรทัดที่ 11

บรรทัดที่ 7 คำสั่งรีเพททิทีฟนี้ จะบวก p เข้ากับ mp ไปเรื่อย ๆ จนกว่า mp จะมีค่ามากกว่า หรือ เท่ากับ m

บรรทัดที่ 8-9 คำสั่งอัลเทอเนทีฟ (Alternative Command) ของซีเอสพีนี้ ประกอบด้วยคำสั่งการ์ดเด็ด (Guarded Command) 2 คำสั่ง โดยจะเลือกคำสั่งการ์ดเด็ดที่เป็นจริงเพียง 1 คำสั่ง และ ทำงานตามคำสั่งที่อยู่หลังเครื่องหมาย "->" ของคำสั่งการ์ดเด็ดนั้น ๆ

บรรทัดที่ 8 ถ้าค่าของ m เท่ากับ mp จะไม่ทำอะไร (skip)

บรรทัดที่ 9 ถ้าค่าของ m น้อยกว่า mp จะส่งค่าข้อมูลในตัวแปรชื่อ m ให้โปรเซสชื่อ SIEVE ที่มี subscript ถัดจากโปรเซสนี้ โดยจะรอจนกว่าโปรเซส SIEVE นั้น ๆ พร้อมจะรับค่าข้อมูลจากโปรเซสนี้

บรรทัดที่ 11 เครื่องหมาย || ใช้คั่นระหว่างโปรเซสซึ่งทำงานจวบกัน

บรรทัดที่ 11-12 ระบุโปรเซสชื่อ SIEVE ที่มี subscript เป็น 0 ประกอบด้วยชุดของคำสั่งหลังเครื่องหมาย ::

บรรทัดที่ 11 จะส่งค่า 2 ให้โปรเซสชื่อ print โดยจะรอจนกว่าโปรเซส print พร้อมจะรับข้อมูลจากโปรเซสนี้ และ กำหนดตัวแปรชนิดเลขจำนวนเต็ม n ให้มีค่าเท่ากับ 3

บรรทัดที่ 12 คำสั่งรีเพททิทีฟนี้ จะส่งค่าของ n ให้โปรเซสชื่อ SIEVE(1) และเพิ่มค่าของ n ขึ้นอีก 2 จะทำเช่นนี้ไปจนกว่าค่าของ n มากกว่าหรือเท่ากับ 10000

บรรทัดที่ 13 ระบุโปรเซสชื่อ SIEVE มี subscript เป็น 101 ประกอบด้วยคำสั่งรีเพททิทีฟซึ่งกำหนดตัวแปร n เป็นชนิดจำนวนเต็ม และรับค่าจากโปรเซส SIEVE(100) มาเก็บในตัวแปร n และ ส่งค่าของ n นี้ให้โปรเซส print จะทำเช่นนี้ไปจนกว่าโปรเซส SIEVE(100) หยุดการประมวลผล

บรรทัดที่ 14 ระบุโปรเซสชื่อ print ประกอบด้วยคำสั่งรีเพททิทีฟ ซึ่งตัวแปร n เป็นชนิดเลขจำนวนเต็ม และ รับค่าจากโปรเซส SIEVE ที่มี subscript เป็น 0 ถึง 101 ที่พร้อมจะส่งข้อมูลมา ค่าที่ได้เก็บไว้ที่ n และ ทำคำสั่งหลังเครื่องหมาย "->" จะทำคำสั่งนี้จนกว่าไม่มีโปรเซส SIEVE ใดส่งข้อมูลมาให้ (คำสั่งหลังเครื่องหมาย "->" นี้ควรเป็นการแสดงผล แต่ในที่นี้ไม่สำคัญสำหรับโครงสร้างของโปรแกรม จึงมิได้ระบุไว้ว่าเป็นอะไร โดยเพียงใส่เครื่องหมายละเอาไว้)

ประเภทของคำสั่งปฐมแบบซีเอสพี

คำสั่งปฐมแบบซีเอสพี มี 7 คำสั่ง คือ คำสั่งพาราเลล (Parallel Command), คำสั่งรีเพททิทีฟ (Repetitive Command), คำสั่งอัลเทอเนทีฟ (Alternative Command), คำสั่งการ์ดเด็ด (Guarded Command), คำสั่งอินพุท (Input Command), คำสั่งเอาท์พุท (Output Command) และ คำสั่งกำหนดค่า (Assignment Command)

1. คำสั่งพาราเลล (Hoare, C.A.R., 1978)

เป็นคำสั่งที่กำหนดการประมวลผลแบบจวบกันของโปรเซสต่าง ๆ ที่ระบุ คำสั่งพาราเลลจะเริ่มประมวลผลโปรเซสต่าง ๆ เหล่านั้น และ จะจบการประมวลผลคำสั่งเมื่อโปรเซสเหล่านั้นจบการประมวลผลลง โปรเซสต่าง ๆ ที่ประมวลผลแบบจวบกันนั้นจะแยกการประมวลผลออกจากกันโดยเด็ดขาด นั่นหมายถึง ไม่มีการใช้ตัวแปรร่วมกัน

การระบุชื่อโปรเซสที่จะประมวลผลแบบจวบกันจะมี subscript หรือไม่ก็ได้ ในกรณีที่มี subscript ยังสามารถระบุเป็นช่วงของ subscript ได้ ตัวอย่างเช่น subscript ที่มีค่าเป็น i:1..10 หมายความว่า มี subscript ของโปรเซสแรกเป็น 1, subscript ของโปรเซสที่สองเป็น 2, และ เป็นเช่นนี้จนถึงโปรเซสสุดท้ายมี subscript เป็น 10

2. คำสั่งรีเพททิทีฟ (Hoare, C.A.R., 1978)

เป็นคำสั่งกำหนดการทำซ้ำของคำสั่งอัลเทอเนทีฟที่อยู่ภายใน การทำซ้ำจะถูกกระทำไปจนกว่าทุก ๆ การ์ดในคำสั่งอัลเทอเนทีฟเป็นเท็จ

3. คำสั่งอัลเทอเนทีฟ (Hoare, C.A.R., 1978)

แต่ละคำสั่งจะมีคำสั่งการ์ดเด็ดอยู่ภายใน หลักการทำงานของคำสั่งอัลเทอเนทีฟคือ จะเลือกทำงานคำสั่งการ์ดเด็ดที่มีการ์ดเป็นจริงเพียงคำสั่งเดียว เมื่อทำงานในคำสั่งการ์ดเด็ดที่เลือกแล้ว จะเริ่มประมวลผลคำสั่งรีเพททิทีฟที่ครอบคลุมอยู่อีกครั้ง ส่วนในกรณีที่การ์ดของทุก ๆ คำสั่งการ์ดเด็ดภายในเป็นเท็จ คำสั่งอัลเทอเนทีฟนี้จะเป็นเท็จซึ่งมีผลให้คำสั่งรีเพททิทีฟที่ครอบคลุมอยู่จบการประมวลผล

ในบางกรณี อาจมีการ์ดที่เป็นจริงในขณะเดียวกันมากกว่า 1 การ์ด การเลือกทำงานการ์ดใดการ์ดหนึ่งนั้นไม่สามารถทราบได้อย่างแน่ชัด ซึ่งต่างไปจากแนวความคิดของโครงสร้างการเลือก การที่ไม่สามารถทราบว่าการ์ดที่เป็นจริงการ์ดใดจะถูกเลือกประมวล

ผลนั้น เรียกว่า nondeterminism

4. <u>คำสั่งการ์ดเด็ด</u> (Hoare, C.A.R., 1978)

ภายในคำสั่งการ์ดเด็ดแบ่งออกเป็น 2 ส่วน คือ ส่วนที่เป็นการ์ด ซึ่งเปรียบเสมือนเป็นเงื่อนไขว่าจะประมวลผลคำสั่งการ์ดเด็ดนั้น ๆ หรือไม่ และ ส่วนที่เป็นคำสั่งต่าง ๆ ที่จะประมวลผลในกรณีที่เลือกประมวลผลคำสั่งการ์ดเด็ดนั้น ๆ

ส่วนที่เป็นการ์ดอาจจะประกอบด้วย

4.1 นิพจน์แบบบูล (Boolean expression) ซึ่งอาจมีการประกาศตัวแปร (ซึ่งจะมีขอบเขตภายในคำสั่งการ์ดเด็ดนั้น ๆ เท่านั้น) หรือ

4.2 คำสั่งอินพุท หรือ

4.3 นิพจน์แบบบูล ตามด้วยคำสั่งอินพุท

ซึ่งจะเรียก 2 ประเภทหลังว่าเป็น "อินพุทการ์ด" (Input Guard) การ์ดประเภทแรกจะเป็นจริงเมื่อนิพจน์ที่ระบุเป็นจริง ส่วนการ์ดประเภทที่สองจะเป็นจริงต่อเมื่อมีคำสั่งเอาท์พุทที่สอดคล้องกันรออยู่ (คำว่า "สอดคล้องกัน" หมายความว่า โปรเซสที่ประมวลผลคำสั่งอินพุทต้องเป็นโปรเซสเดียวกับที่ระบุในคำสั่งเอาท์พุท และ โปรเซสที่ประมวลผลคำสั่งเอาท์พุทต้องเป็นโปรเซสเดียวกับที่ระบุในคำสั่งอินพุท) สำหรับการ์ดประเภทสุดท้ายจะเป็นจริงต่อเมื่อทั้ง 2 กรณีเป็นจริง

5. <u>คำสั่งอินพุท</u> (Hoare, C.A.R., 1978)

เป็นคำสั่งหนึ่งที่ใช้ในการติดต่อระหว่างโปรเซส 2 โปรเซสที่ประมวลผลจบกัน โดยจะเป็นคำสั่งที่ใช้รอรับค่าข้อมูลจากโปรเซสที่ต้องการ ซึ่งคำสั่งอินพุทต้องระบุชื่อโปรเซสที่จะส่งค่ามาให้ นอกจากนี้ คำสั่งอินพุทต้องระบุตัวแปรที่จะใช้รอรับค่าด้วย

ในกรณีที่ โปรเซสต้นทางที่ระบุในคำสั่งอินพุทยังไม่ประมวลผลคำสั่งเอาท์พุทที่มีชื่อโปรเซสนี้เป็นโปรเซสปลายทาง คำสั่งอินพุทนี้จะต้องรอจนกว่าจะมีการประมวลผลคำสั่งเอาท์พุทที่สอดคล้องกันจากโปรเซสต้นทาง จึงจะสามารถจบการประมวลผลคำสั่งนี้และทำงานคำสั่งต่อไปได้ แต่ถ้าหากว่า โปรเซสต้นทางที่ระบุนั้นจบการประมวลผลไปแล้ว หรือ มิได้ประมวลผลอยู่ คำสั่งอินพุทนี้จะจบการประมวลผลลงทันทีและตัวแปรที่ถูกกำหนดให้รับค่าจะมิได้รับค่าที่ต้องการ

6. <u>คำสั่งเอาท์พุท</u> (Hoare, C.A.R., 1978)

เป็นอีกคำสั่งหนึ่งที่ใช้ในการติดต่อระหว่างโปรเซส 2 โปรเซสที่ประมวลผลจบ

กัน โดยจะเป็นคำสั่งที่ใช้ส่งค่าข้อมูลให้โปรเซสที่ต้องการซึ่งคำสั่งเอาท์พุทต้องระบุชื่อโปรเซสที่จะรอรับค่าให้ นอกจากนี้ คำสั่งเอาท์พุทต้องระบุตัวแปรที่จะใช้ส่งค่าให้ด้วย

ในกรณีที่โปรเซสปลายทางที่ระบุในคำสั่งเอาท์พุทยังไม่ประมวลผลคำสั่งอินพุทที่มีชื่อโปรเซสนี้เป็นโปรเซสต้นทาง คำสั่งเอาท์พุทนี้จะต้องรอจนกว่าจะมีการประมวลผลคำสั่งอินพุทที่สอดคล้องกันจากโปรเซสปลายทาง จึงจะสามารถจบการประมวลผลคำสั่งนี้และทำงานคำสั่งต่อไปได้ แต่ถ้าหากว่า โปรเซสปลายทางที่ระบุนั้นจบการประมวลผลไปแล้ว หรือ มิได้ประมวลผลอยู่ คำสั่งเอาท์พุทนี้จะจบการประมวลผลลงทันที

7. <u>คำสั่งกำหนดค่า</u> (Hoare, C.A.R., 1978)

เป็นคำสั่งที่กำหนดค่าของนิพจน์ที่ระบุ (ซึ่งอยู่ทางขวามือของคำสั่ง) ให้กับตัวแปรที่ระบุ (ซึ่งอยู่ทางซ้ายมือของคำสั่ง)