# บทที่ 3

## การออกแบบและพัฒนา

ระบบที่ใช้ในการออกแบบและพัฒนาโปรแกรมประกอบด้วย 2 ระบบคือ

- บนระบบปฏิบัติการไมโครซอฟท์วินโดส์ 95 (MS Windows 95)
  อุปกรณ์ที่ใช้คือ เครื่องคอมพิวเตอร์ใช้หน่วยประมวลผล
  อินเทลเพนเทียม 100 เมกะเฮิรทซ์ (Intel Pentium 100 MHz)
  หน่วยความจำหลักขนาด 40 เมกะไบต์ และโมเด็ม (MODEM)
  ขนาดความเร็ว 28.8 กิโลบิตต่อวินาที โดยติดต่อ
  กับระบบเครือข่ายด้วย Point to Point Protocol (PPP)
  เครื่องมือที่ใช้ในการพัฒนาโปรแกรมคือ
  ไมโครซอฟท์วิชวลเจพลัสพลัส 1.1 (MS Visual J++ 1.1),
  ฮอทจาวาเบราเซอร์ 1.1.2 สำหรับวินโดส์ 95 (Hotjava Browser 1.1.2 for Windows 95)
  และเจดีเค 1.1 สำหรับวินโดส์ 95 (JDK 1.1 for Windows 95)
- บนระบบปฏิบัติการซันโซลาริส 2.5.1 (SUN Solaris 2.5.1)
  อุปกรณ์ที่ใช้คือ เครื่องคอมพิวเตอร์ใช้หน่วยประมวลผล
  ซันอุลตร้าสปาร์ค 143 เมกะเฮิรทซ์ (SUN Ultra SPARC 143 MHz)
  หน่วยความจำหลักขนาด 96 เมกะไบต์และติดต่อกับระบบเครือข่ายด้วย Ethernet
  ความเร็ว 10 เมกะบิตต่อวินาที เครื่องมือที่ใช้ในการพัฒนาโปรแกรมคือเจดีเค 1.1
  สำหรับโซลาริส (JDK 1.1 for Solaris) และโปรแกรมฮอทจาวาเบราเซอร์ 1.1.2
  สำหรับโซลาริส (Hotjava Browser 1.1.2 for Solaris)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



การออกแบบและพัฒนาโปรแกรมแสดงสถานะเครือข่ายแบ่งออกเป็น 3 ส่วนด้วยกันคือ

1. พัฒนาส่วนจัดการติดต่อกับตัวแทนส่วนนี้จะทำหน้าที่ในการรับและส่งเม็สเสจ SNMP กับตัวแทน
2. พัฒนาส่วนจัดการสำรวจ (probe) เครือข่ายซึ่งจะทำหน้าที่สำรวจระบบเครือข่ายว่าประกอบด้วยอุปกรณ์อะไรบ้าง ทั้งนี้อุปกรณ์เหล่านี้จะต้องสนับสนุน SNMP
3. ส่วนตัวจัดการจะทำหน้าที่แสดงข้อมูลที่ได้รับจากตัวแทนในลักษณะที่ผู้บริหารเครือข่ายสามารถนำไปใช้ประโยชน์ได้

## 3.1 การพัฒนาส่วนจัดการติดต่อกับตัวแทน

เม็สเสจที่ส่วนนี้จะต้องสร้างขึ้นคือ

- GetRequest
- GetNextRequest
- GetResponse

โดยที่แต่ละเม็สเสจจะมีรูปแบบดังนี้

| PDU type | request-id | 0 | 0 | variable-bindings |
|---|---|---|---|---|

GetRequest PDU, GetNextRequest PDU

| PDU type | Requeat-id | error-status | error-index | variable-bindings |
|---|---|---|---|---|

GetResponse PDU

รูปที่ 3.1 รูปแบบของ GetRequest PDU, GetNextRequest PDU และ GetResponse PDU

ใน GetRequest และ GetNextRequest ส่วนของ variable-binding นั้นค่าของ OBJECT IDENTIFIER จะถูกกำหนดให้เป็น null หรือ 0 เมื่อตัวแทนได้รับเม็สเสจก็จะส่ง GetResponse กลับมาโดย OBJECT IDENTIFIER ยังเป็นตัวเดิมแต่ค่าของมันจะเป็นค่าที่เก็บอยู่ในตัวแทน

ส่วนของ variable-binding ประกอบด้วย 2 ส่วนคือ

- OBJECT IDENTIFIER
- ค่าของ OBJECT IDENTIFIER

| ชนิดข้อมูล | ค่าที่กำหนดในรูปของเลขฐาน 16 |
|---|---|
| INTEGER | 02 |
| BIT STRING | 03 |
| OCTET STRING | 04 |
| NULL | 05 |
| OBJECT IDENTIFIER | 06 |
| SEQUENCE | 30 |
| IpAddress | 40 |
| Counter | 41 |
| Gauge | 42 |
| TimeTicks | 43 |
| Opaque | 44 |
| NsapAddress | 45 |
| Counter64 | 46 |
| Counter32 | 47 |
| GetRequest-PDU | A0 |
| GetNextRequest-PDU | A1 |
| GetResponse-PDU | A2 |
| SetRequest-PDU | A3 |
| Trap-PDU | A4 |

ตารางที่ 3.1 แสดงชนิดข้อมูลของ OBJECT IDENTIFIER ที่ใช้ใน SNMP

ในการสร้างเม็สเสจนั้นข้อมูลทุกตัวจะถูกเข้ารหัสโดยใช้หลักการของ Abstract Syntax Notation 1 (ASN.1) variable-binding ที่ถูกเข้ารหัสโดย ASN.1 จะประกอบด้วยส่วนต่างๆ ดังนี้

- ชนิดข้อมูลของ OBJECT IDENTIFIER มีขนาด 1 ไบต์ (byte)
- ขนาดของข้อมูล ในการเก็บขนาดข้อมูลถ้าข้อมูลมีความยาวไม่เกิน 127 ไบต์ ก็จะใช้เนื้อที่ในการเก็บ 1 ไบต์ เช่น

| 04 | 06 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|

ชนิดข้อมูลคือ 04 ซึ่งก็คือ OCTET STRING และมีความยาว 6 ซึ่งไม่เกิน 127 ดังนั้นจะใช้เนื้อที่ในการเก็บขนาดของข้อมูล 1 ไบต์ แต่ถ้าขนาดของข้อมูลยาวกว่า 127 ไบต์ แล้วไบต์แรกที่เก็บความยาวของข้อมูลบิต (bit) ที่ 8 จะถูกกำหนดให้เป็น 1 ส่วนบิตที่เหลือจะถูกใช้ในการบอกขนาด เช่น

| 82 | 01 | 04 |
|---|---|---|

บิตที่ 8 ของไบต์แรกเป็น 1 แสดงว่าขนาดของข้อมูลเกิน 127 ไบต์ ดังนั้นบิตที่ 1 ถึง 7 จะบอกจำนวนไบต์ที่เก็บความยาวของข้อมูล ซึ่งแสดงในรูปของเลขฐาน 2 ได้ 000 0010 ดังนั้นความยาวคือ 2 และจะใช้ 2 ออคเทต (octet) ถัดไปเป็นความยาวของข้อมูล ซึ่งก็คือ 0000 00001 0000 0100 หรือ 132

และข้อมูล OCTET STRING ที่ประกอบด้วยอักษร A จำนวน 132 ตัวจะเป็น

| 04 | 82 | 01 | 04 | A | ... | A |
|---|---|---|---|---|---|---|

- ตัวข้อมูลซึ่งประกอบไปด้วยออคเทตจำนวนหนึ่ง

ในการสร้างเม็สเสจ SNMP นั้นจะถือว่าตัวเม็สเสจก็เป็นข้อมูลตัวหนึ่งที่จะต้องทำการเข้ารหัสด้วยเหมือนกัน โดยมีชนิดข้อมูลเป็น 30 หรือ SEQUENCE ที่จะต้องประกอบด้วย 3 ส่วนคือ ชนิดข้อมูล, ขนาดของข้อมูลและตัวข้อมูล ดังรูปที่ 3.2

| 30 | ขนาดของข้อมูล | ข้อมูล |
|---|---|---|

รูปที่ 3.2 เม็สเสจ SNMP

ดังนั้น GetRequest ที่ส่งไปถามข้อมูล sysUpTime หรือ 1.3.6.1.2.1.1.3.0 โดยที่ระหัสผ่านเป็น "public" จะเป็นดังนี้

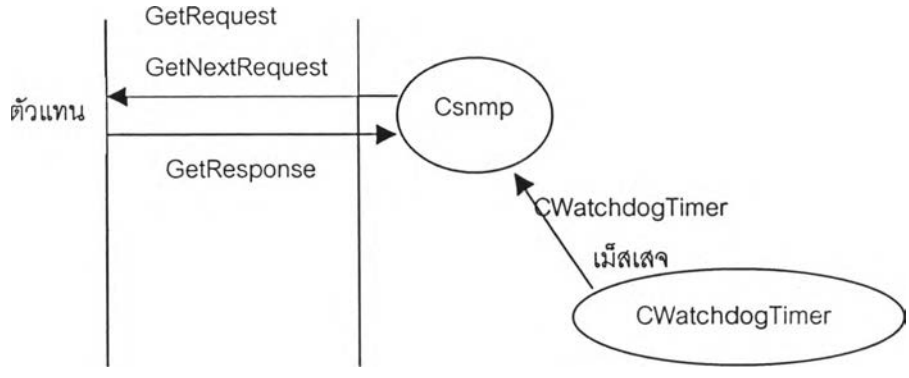
30 24 02 01 00 04 06 70 75 62 6C 69 63 A0 17 02
01 00 02 01 00 02 01 00 30 0C 06 08 2B 06 01 02
01 01 03 00 05 00

โดยที่ 70 75 62 6C 69 63 ก็คือ OCTET STRING ของคำว่า "public" ในการเข้ารหัส OBJECT IDENTIFIER นั้น 1.3 จะเข้ารหัสให้เป็น 2B เพียง byte เดียว ดังนั้น 1.3.6.1.2.1.1.3.0 จะถูกเข้ารหัสให้เป็น 2B 06 01 02 01 01 03 00

การจะติดต่อกับตัวแทนนั้นจะใช้ UDP ซึ่งจะติดต่อกันผ่านพอร์ตหมายเลข 161 บนอุปกรณ์ที่ตัวแทนทำงานอยู่ หากไม่ได้รับเม็สเสจตอบกลับจากตัวแทนภายใน 3 วินาทีโปรแกรมจะทำการส่งเม็สเสจซ้ำอีก หากไม่ได้รับคำตอบหลังจากส่งเม็สเสจซ้ำ 3 ครั้งจะถือว่าตัวแทนไม่ทำงานหรืออุปกรณ์ที่ติดต่อด้วยนั้นไม่สนับสนุน SNMP

ในการพัฒนาโปรแกรมส่วนนี้ประกอบด้วย 3 คลาส (class) คือ

1. Csnmp เป็นคลาสที่ทำหน้าที่ในการสร้าง SNMP เม็สเสจขึ้นมาและจะเป็นคลาสที่ให้บริการติดต่อกับตัวแทนให้กับคลาสอื่นๆ อีกที method ที่คลาสนี้ให้บริการแก่คลาสอื่นคือ
   - getRequest ส่ง GetRequest ไปให้กับตัวแทน
   - getNextRequest ส่ง GetNextRequest ไปให้กับตัวแทน
   - getRequest ส่ง CsnmpData กลับไปให้กับคลาสที่เรียกใช้มัน
2. CWatchdogTimer เป็นคลาสที่ใช้ในการจับเวลาและส่งเม็สเสจให้กับคลาสที่เรียกใช้มันเพื่อบอกว่าหมดเวลาแล้ว
3. CsnmpData เป็นคลาสที่ใช้เก็บข้อมูลที่ได้จากการทำงานของ Csnmp ซึ่งประกอบด้วย OBJECT IDENTIFIER, ค่าของ OBJECT IDENTIFIER และข้อผิดพลาด(error) ที่เกิดจากการติดต่อกับตัวแทน




รูปที่ 3.3 แสดงความสัมพันธ์ของ คลาส ในการพัฒนาโปรแกรมในส่วนที่ 1

จากรูปที่ 3.3 การทำงานในส่วนนี้จะเริ่มจากคลาส Csnmp ส่งเม็สเสจไปขอข้อมูลกับตัวแทนและในขณะเดียวกันคลาส Csnmp ก็เรียกคลาส CWatchdogTimer ขึ้นมาทำงานด้วยโดย CWatchdogTimer จะคอยจนครบเวลาที่ตั้งไว้โดยผู้พัฒนาตั้งไว้ที่ 5 วินาที การทำงานจะสิ้นสุดเมื่อมีเม็สเสจถูกส่งมาให้กับ Csnmp ไม่ว่าจะเป็น GetResponse หรือ CWatchdogTimer ถ้าเม็สเสจที่ได้รับเป็น GetResponse ถือว่าตัวแทนทำงานตามปกติ แต่ถ้าหากเม็สเสจที่ได้รับเป็น CwatchdogTimer เม็สเสจแสดงว่าเกิดความผิดพลาดขึ้นในการส่งเม็สเสจไปยังตัวแทนซึ่งอาจจะเกิดจากปัญหาของระบบเครือข่ายหรือปัญหาที่เกิดขึ้นกับอุปกรณ์ที่ตัวแทนทำงานอยู่ก็ได้

หากไม่ได้รับเม็สเสจส่งกลับมาจากตัวแทน Csnmp จะส่งเม็สเสจขอข้อมูลไปที่ตัวแทนอีกครั้ง หากทำครบ 3 ครั้งแล้ว เม็สเสจที่ได้รับยังเป็น CwatchdogTimer เม็สเสจ จะถือว่าไม่สามารถติดต่อกับตัวแทนได้

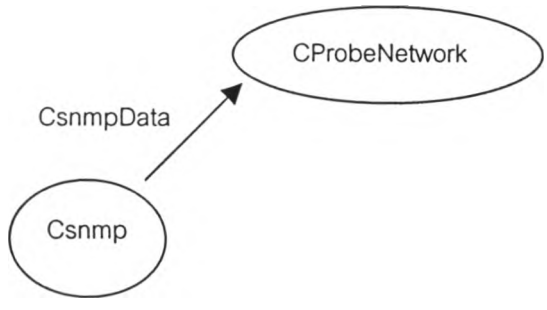
### 3.2 การพัฒนาโปรแกรมส่วนของการสำรวจ (probe) เครือข่าย

การพัฒนาโปรแกรมในส่วนนี้ทำโดยการกระจายเม็สเสจ SNMP ให้กับทุกอุปกรณ์ที่อยู่ภายในระบบเครือข่าย เช่น เครือข่าย 161.200.93.0 ซึ่งมีเน็ตมาส (netmask) เป็น 255.255.255.0 และบรอดคาสแอดเครส (boardcast address) เป็น 161.200.93.255 หากส่งเม็สเสจไปที่บรอดคาสแอดเดรสนี้ ส่วนตัวแทนที่ทำงานอยู่บนอุปกรณ์ต่างๆ ในระบบเครือข่ายก็จะส่ง GetResponse กลับมา ทำให้ทราบได้ว่าอุปกรณ์ใดบ้างในระบบเครือข่ายที่มีตัวแทนทำงานอยู่

การพัฒนาโปรแกรมในส่วนนี้ประกอบด้วยคลาสดังนี้

1 CProbeNetwork ทำหน้าที่ส่งเม็สเสจไปยังบรอดคาสแอดเดรสของระบบเครือข่าย และคอยรับเม็สเสจที่ส่งกลับมาจากตัวแทนภายในระบบเครือข่ายนั้น โดยใช้บริการของคลาส Csnmp

2 CHostList เป็นคลาสที่ใช้เก็บจำนวนของอุปกรณ์และรายชื่อของอุปกรณ์ที่ได้จากการ probe




รูปที่ 3.4 แสดงความสัมพันธ์ของคลาสในการพัฒนาโปรแกรมในส่วนที่ 2

## 3.3 ส่วน ตัวจัดการ

การทำงานของโปรแกรมแสดงสถานะเครือข่ายในส่วนนี้จะเป็นส่วนที่จะนำข้อมูลต่างๆ ที่อยู่ในแต่ละอุปกรณ์มาใช้ในการช่วยให้ผู้บริหารเครือข่ายทราบถึงสถานะของเครือข่ายโดยจะแสดงข้อมูลต่อไปนี้

### 3.3.1 รายละเอียดของอุปกรณ์

แสดงข้อมูลทั่วๆ ไปของอุปกรณ์ คือ

- sysDescr รายละเอียดของระบบ
- sysContact ข้อมูลบุคคลที่จะติดต่อ
- sysName ชื่อของอุปกรณ์
- sysLocation สถานที่ที่ตั้งของอุปกรณ์

### 3.3.2 รายละเอียดของแต่ละอินเตอร์เฟสของอุปกรณ์

แสดงข้อมูลของอินเตอร์เฟสของอุปกรณ์คือ

- ifDescr แสดงรายละเอียดของอินเตอร์เฟส
- ifType ประเภทของอินเตอร์เฟส
- ifMtu แสดง MTU ของอินเตอร์เฟส
- ifSpeed ความเร็วของอินเตอร์เฟส
- ifPhysAddress แสดงฟิสิคอลแอดเดรสของอินเตอร์เฟส
- ifAdminStatus แสดงสถานะของอินเตอร์เฟสซึ่งประกอบด้วย up, down, testing
- ifInOctets ปริมาณข้อมูลที่ไหลเข้าอินเตอร์เฟส
- ifOutOctets ปริมาณข้อมูลที่ไหลออกจากอินเตอร์เฟส
- ipAdEntAddr หมายเลข IP ที่กำหนดให้กับอินเตอร์เฟส
- ipAdEnNetMask แสดงเน็ตมาสที่กำหนดให้กับอินเตอร์เฟส
- ipAdEntBcastAddr แสดงบรอดคาสแอดเดรสของอินเตอร์เฟส

### 3.3.3 แสดงตารางการหาเส้นทาง (routing table) ของอุปกรณ์

ประกอบด้วย

- ipRouteDest แสดงเป้าหมาย (destination) ของเส้นทาง (route)
- ipRouteIfIndex หมายเลขอินเตอร์เฟสสำหรับเส้นทาง
- ipRouteNextHop แสดงจุดถัดไปที่จะทำการส่งข้อมูลออกไปตามเส้นทางนี้
- ipRouteMask แสดงเน็ตมาสของเส้นทาง

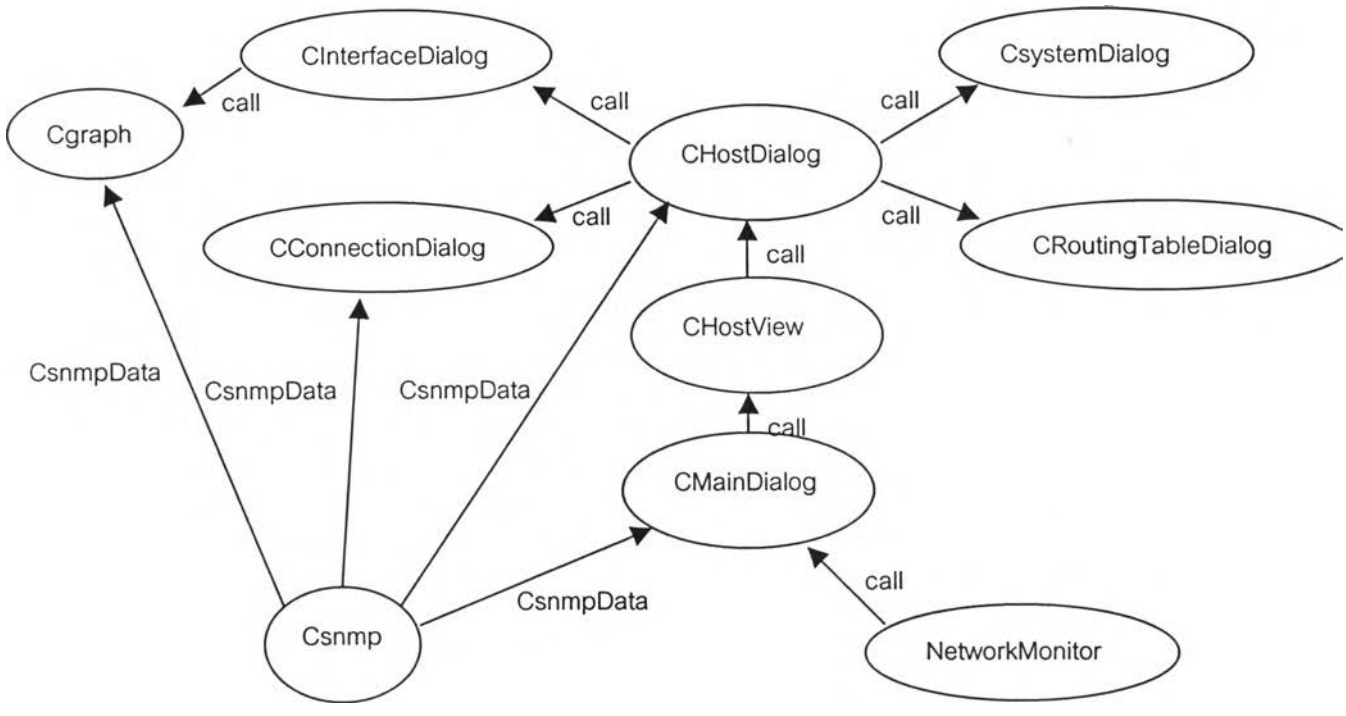
### 3.3.4 แสดงการติดต่อระหว่างอุปกรณ์กับอุปกรณ์อื่นๆ ในเครือข่ายที่กำลังเกิดขึ้น

จะแสดงข้อมูลการติดต่อกับอุปกรณ์ต่างๆ ในระดับ TCP ดังนี้

- tcpConnState แสดงสถานะของการติดต่อ (connection)
- tcpConnLocal แสดงหมายเลข IP ของการติดต่อบนอุปกรณ์
- tcpConnLocalPort หมายเลขพอร์ตของการติดต่อบนอุปกรณ์
- tcpConnRem แสดงหมายเลข IP ของการติดต่อบนอุปกรณ์ที่ติดต่อด้วย
- tcpConnRemPort หมายเลขพอร์ตของการติดต่อบนอุปกรณ์ที่ติดต่อด้วย

การพัฒนาโปรแกรมในส่วนนี้ประกอบด้วยคลาสต่างๆ ต่อไปนี้

1. NetworkMonitor เป็นคลาสหลักที่ทำหน้าที่รับข้อมูลบรอดคาสแอดเดรสและระหัสผ่านจากผู้ใช้ และทำการเรียกคลาส CmainDialog ขึ้นมาทำงาน
2. CMainDialog เป็นคลาสที่เรียกใช้คลาส CprobeNetwork เพื่อเก็บรวบรวมรายชื่อของอุปกรณ์ต่างๆ ในระบบเครือข่ายที่มีตัวแทนทำงานอยู่ และทำการเรียกคลาส ChostView ขึ้นมาทำการแสดงอุปกรณ์ต่างๆ เป็นรูปภาพอีกที
3. CHostView เป็นคลาสที่ใช้ในแสดงอุปกรณ์ต่างๆ ในระบบเครือข่ายด้วยรูปภาพและคอยรับคำสั่งจากผู้ใช้โปรแกรมว่าต้องการให้แสดงรายละเอียดของอุปกรณ์ใด
4. CHostDialog เป็นคลาสที่แสดงหน้าต่างรายละเอียดของอุปกรณ์ และให้ผู้ใช้โปรแกรมเลือกว่าจะให้โปรแกรมแสดงรายละเอียดของอุปกรณ์กลุ่มใดในกลุ่มระบบ, กลุ่มอินเตอร์เฟส, กลุ่มตารางเส้นทาง หรือกลุ่มตารางแสดงการติดต่อด้วยโพรโตคอล TCP
5. CSystemDialog ทำหน้าที่แสดงข้อมูลระบบของอุปกรณ์
6. CInterfaceDialog ทำหน้าที่แสดงข้อมูลที่เกี่ยวกับอินเตอร์เฟสของอุปกรณ์
7. CRoutingTableDialog ทำหน้าที่แสดงข้อมูลที่เกี่ยวกับตารางเส้นทางของอุปกรณ์
8. CConnectionDialog ทำหน้าที่แสดงข้อมูลที่เกี่ยวกับตารางการติดต่อของอุปกรณ์
9. CGraph ทำหน้าที่แสดงกราฟอรรถประโยชน์ (utilization) ของอินเตอร์เฟสของอุปกรณ์




รูปที่ 3.5 แสดงความสัมพันธ์ของคลาสในการพัฒนาโปรแกรมในส่วนที่ 3