

บราวเซอร์ที่สามารถโปรแกรมได้

Programmable Browser



เลขหมึก.....  
เลขทะเบียน... 42789  
วัน, เดือน, ปี 10 ส.ย. 2545

.b.....
.i.....

ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2543

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง บราวเซอร์ที่สามารถโปรแกรมได้

Programmable Browser

ผู้จัดทำ

1. นาย วชิระ ศิริพจนาวรรณ รหัสประจำตัว 40010660
2. นางสาว สุพัฒน์ดา โชติพันธ์ รหัสประจำตัว 40010892



อาจารย์ที่ปรึกษา

(ดร. วิศิษฐ์ ทรัพย์กิตติ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บราวเซอร์ที่สามารถโปรแกรมได้

นายวชิระ ศิริพจนาวรรณ รหัส 40010660  
นางสาวสุพัฒน์ดา โชติพันธ์ รหัส 40010892  
ดร.วิศิษฎ์ หิรัญกิตติ อาจารย์ที่ปรึกษา  
ปีการศึกษา 2543

## บทคัดย่อ

ปฏิญานិพนธ์ฉบับนี้เป็นการพัฒนาโปรแกรมเว็บบราวเซอร์ให้สามารถโปรแกรมได้ตามความต้องการของผู้ใช้ บราวเซอร์ที่สามารถโปรแกรมได้เป็นโปรแกรมเว็บบราวเซอร์ที่มีความยืดหยุ่น อนุญาตให้ผู้ใช้ท่องเว็บได้เหมือนกับที่บราวเซอร์อื่น ๆ ทำได้ นอกจากนี้ยังอนุญาตให้ผู้ใช้ควบคุมการทำงานต่างๆ ของโปรแกรมได้ด้วยภาษาสคริปต์ที่ผู้ใช้เขียนขึ้นเอง

ผู้ใช้สามารถสั่งให้บราวเซอร์ทำงานต่าง ๆ ได้โดยอัตโนมัติ เช่น เปิด-ปิดโปรแกรมตามเวลาที่กำหนด เป็นต้น บราวเซอร์ที่สามารถโปรแกรมได้สามารถติดต่อไปยังนิวส์เซิร์ฟเวอร์ที่ระบุไว้เพื่อร้องขอบทความต่าง ๆ จากกลุ่มข่าวที่ระบุไว้ ดาวน์โหลดบทความเหล่านั้น มาเก็บไว้บนเครื่องคอมพิวเตอร์ และส่งบทความต่าง ๆ ไปให้กับเพื่อนได้โดยอัตโนมัติ ผู้ใช้สามารถเขียนสคริปต์หรืออ่านไฟล์สคริปต์ขึ้นมาทำงานเพื่อตอบสนองความต้องการของตนเองได้ และสามารถบันทึกไฟล์สคริปต์เก็บไว้ใช้งานได้ นอกจากนี้ สคริปต์ยังมีความสามารถพิเศษอื่น ๆ อีกที่สามารถขยายเพิ่มเติมได้อย่างไม่จำกัด ตามความสามารถของโปรแกรมที่ถูกเรียกใช้งานจากเครื่องเซิร์ฟเวอร์ เช่น ค้นหาที่ต้องการจากเครือข่ายอินเทอร์เน็ต หรือส่งข้อความไปยังเพจเจอร์ เป็นต้น

นอกจากนี้ผู้ใช้อังสามารถใช้โปรแกรมเหล่านี้เพื่อติดต่อไปยังโปรแกรมบราวเซอร์ที่สามารถโปรแกรมนี้ได้และสามารถสั่งให้ทำงานได้จากระยะไกลผ่านโปรแกรมเหล่านี้

## Programmable Browser

Wachira Siripochanawan

Supannada Chotipant

Dr. Visit Hirankitti Advisor

### Abstract

This thesis describes the development of a web browser, which is programmable, or so-called “Programmable Browser”. Programmable Browser is a flexible web browser that allows the user to browse the web like any web browser can do. Additionally, Programmable Browser allows the user to program its activities by scripts.

The scripts provide the browser’s automatic activities, for example, user can specify in the script to open and close the browser at specific time. Programmable Browser can connect to a specific news server, request the articles from a specific news group and download them to the personal computer and finally automatically send them to his/her friends. The user can program or load his/her own scripts for his/her own purposes. The user can store his/her scripts for future use. The script’s functionality can be extended with no limit according to the functionality that can be provided by a web (CGI) application. For example, the user can program the script to search for keywords in Internet and then send the messages to a mobile telephone.

Finally, you can use program Telnet to connect to this Programmable Browser and control it remotely machine from a long distance.

## กิตติกรรมประกาศ

หลายสิ่งหลายอย่างที่คณะผู้จัดทำคาดไม่ถึงได้เกิดขึ้นมากมายเมื่อได้เริ่มลงมือเขียนปริญญาานิพนธ์ฉบับนี้ โชคดีที่คณะผู้จัดทำมิได้ถูกทอดทิ้งให้เขียนปริญญาานิพนธ์ฉบับนี้เพียงลำพัง ปริญญาานิพนธ์ฉบับนี้คงไม่เสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือจากบุคคลต่าง ๆ หลายฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงและเป็นบุคคลที่มีส่วนสำคัญต่อความสำเร็จของปริญญาานิพนธ์ฉบับนี้มากที่สุด คือ อาจารย์ ดร.วิศิษฎ์ หิรัญกิตติ อาจารย์ที่ปรึกษาปริญญาานิพนธ์ หลายครั้งที่อาจารย์สละเวลาส่วนตัวซึ่งเป็นเวลาที่อาจารย์ควรพักผ่อนมาให้คำปรึกษา มาคอยดูแลเอาใจใส่ และคอยช่วยเหลือเสมอมา ต้องขอขอบพระคุณเป็นอย่างสูง

นอกจากนี้ต้องขอขอบคุณ ฝ่ายธุรการภาควิชาวิศวกรรมคอมพิวเตอร์, อาจารย์บัณฑิต พัสยา และ อาจารย์ธนา หงษ์สุวรรณ ที่คอยประสานงานอำนวยความสะดวกเป็นอย่างดี ขอขอบคุณอาจารย์ อัครเดช วัชรภพพงษ์ ที่ให้คำแนะนำ, ข้อคิด และความบันเทิง ขอขอบคุณอาจารย์ทุกท่านที่ประสิทธิ์ประสาทความรู้ และขอขอบคุณเพื่อน ๆ ทุกคนในห้อง Intelligent Communication LAB, ห้อง OLALA, ห้อง Hardware, ห้อง Network และห้อง ISAG ที่คอยให้กำลังใจและความบันเทิงอยู่เสมอมา

และต้องขอขอบพระคุณบุคคลที่มีความสำคัญที่สุดที่ทำให้วันนี้ได้ ก็คือ บิดา และ มารดาอันเป็นที่เคารพและรักยิ่งซึ่งได้อบรมเลี้ยงดูมาเป็นอย่างดี พร้อมทั้งได้ให้โอกาสในการศึกษาที่ดีที่สุด และให้กำลังใจในการทำงานเสมอมา คณะผู้จัดทำขอระลึกในพระคุณ และขอกราบขอบพระคุณเป็นอย่างสูงมา ณ ที่นี้

สุพัฒน์ดา โชติพันธ์  
วชิระ สิริพจนาวรรณ

## สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VI
สารบัญรูปภาพ	VII
บทที่ 1 บทนำ	1
1.1 ที่มาของโครงการ	1
1.2 วัตถุประสงค์	1
1.3 เป้าหมาย	2
1.4 ขอบเขตของโครงการ	2
1.5 วิธีการดำเนินงาน	2
บทที่ 2 หลักการพื้นฐาน	3
2.1 หลักการพื้นฐานของเอเจนต์ (Agents)	3
2.2 หลักการพื้นฐานของเว็บเบราว์เซอร์	6
2.3 พื้นฐานต่างๆ เกี่ยวกับเครือข่าย	8
2.4 เอชทีทีพี (HTTP)	20
2.5 หลักการพื้นฐานของอาร์เอ็มไอ (RMI)	25
2.6 หลักการพื้นฐานของเซรค	29
บทที่ 3 แนวคิดหลักในการพัฒนาโปรแกรม	34
3.1 บทนำ	34
3.2 การจัดการคำสั่งสคริปต์	35
3.3 การทำงานของเอเจนต์	39
3.4 การส่งและรับข่าวของเบราว์เซอร์ที่สามารถโปรแกรมได้ผ่าน NNTP	41
3.5 การทำงานของเบราว์เซอร์ที่สามารถโปรแกรมได้ผ่าน โปรแกรมเทลเน็ต	44
บทที่ 4 การพัฒนาโปรแกรม	47
4.1 บทนำ	47
4.2 การจัดการคำสั่งสคริปต์	48
4.3 การทำงานของเอเจนต์	55
4.4 การส่งและรับข่าวของเบราว์เซอร์ที่สามารถโปรแกรมได้ผ่าน NNTP	57
4.5 การทำงานของเบราว์เซอร์ที่สามารถโปรแกรมได้ผ่าน โปรแกรมเทลเน็ต	58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้าที่
บทที่ 5 การทำงานของบราวเซอร์	59
5.1 การใช้งานบราวเซอร์ที่สามารถโปรแกรมได้ด้วยคำสั่งสคริปต์	59
5.2 วิธีการใช้งานบราวเซอร์จากระยะไกลด้วยโปรแกรมเทลเน็ต	66
5.3 การส่งและรับข่าวของบราวเซอร์ที่สามารถโปรแกรมได้ผ่าน NNTP	69
บทที่ 6 บทสรุป	72
6.1 บทสรุปและวิจารณ์	72
6.2 ข้อดีของบราวเซอร์ที่สามารถโปรแกรมได้	72
6.3 ข้อเสียของบราวเซอร์ที่สามารถโปรแกรมได้	72
6.4 แนวทางการพัฒนา	73
ภาคผนวก ก.	74
ภาคผนวก ข.	77
บรรณานุกรม	83



## สารบัญตาราง

	หน้าที่
ตารางที่ 2-1 แสดงคุณสมบัติต่างๆของเอเจนต์	4
ตารางที่ 2-2 ตัวอย่างบราวเซอร์ที่มีในปัจจุบัน	7
ตารางที่ 2-3 แสดงองค์ประกอบต่าง ๆ ของส่วนที่ระบุชื่อรีซอร์ส	11
ตารางที่ 5-1 ตารางแสดงตัวอย่างของ Currency และความหมาย	61



## สารบัญรูปภาพ

หน้าที่

รูปที่ 2-1 แสดงลำดับขั้นของเอเจนต์	5
รูปที่ 2-2 แสดงลักษณะการทำงานของบราวเซอร์	8
รูปที่ 2-3 แสดงเลขอร์ต่าง ๆ	8
รูปที่ 2-4 แสดงการติดต่อระหว่างไคลเอ็นต์และเซิร์ฟเวอร์	10
รูปที่ 2-5 แสดงการติดต่อโดยใช้พร็อกซี	10
รูปที่ 2-6 แสดงตัวอย่างของยูอาร์แอลที่อยู่ของเว็บไซต์จาวา	11
รูปที่ 2-7 แสดงการร้องขอของไคลเอ็นต์ไปยังเซิร์ฟเวอร์	17
รูปที่ 2-8 แสดงการเชื่อมต่อระหว่างไคลเอ็นต์และเซิร์ฟเวอร์	17
รูปที่ 2-9 แสดงการมาร์แชลลิ่งและอันมาแชลลิ่ง	26
รูปที่ 2-10 แสดงการทำงานของรีจิสทรีตามวิธีการรีโมตเมธอดอิน โวเคชัน	27
รูปที่ 2-11 แสดงการติดต่อสื่อสารผ่านสตับและสเกเลตันของวัตถุโลกคอลและวัตถุระยะไกล	29
รูปที่ 2-12 แสดงวัฏจักรของเรียด	31
รูปที่ 3-1 แสดงการทำงานของเอเจนต์	35
รูปที่ 3-2 แสดงการทำงานของภาษาสคริปต์แบบเงื่อนไขเดียว	36
รูปที่ 3-3 แสดงการทำงานของภาษาสคริปต์แบบสองเงื่อนไข	37
รูปที่ 3-4 แสดงการทำงานของภาษาสคริปต์แบบลูป while	37
รูปที่ 3-5 แสดงการทำงานของภาษาสคริปต์แบบลูป repeat-until	38
รูปที่ 3-6 แสดงการติดต่อระหว่างเอเจนต์และเซิร์ฟเวอร์	39
รูปที่ 3-7 แสดงการติดต่อระหว่างเอเจนต์และนิวส์เซิร์ฟเวอร์	42
รูปที่ 3-8 แสดงการประยุกต์ใช้รีโมตเมธอดอิน โวเคชันกับเอเจนต์	42
รูปที่ 3-9 แสดงการส่งงานของบราวเซอร์ผ่านโปรแกรมเทเลเน็ต	45
รูปที่ 4-1 แสดงการเก็บค่าของตัวแปรแบบ Integer และ String	49
รูปที่ 4-2 แสดงการดึงค่าของตัวแปรออกมาใช้งาน	49
รูปที่ 4-3 แสดงการทำงานของคำสั่ง if เมื่อเงื่อนไขเป็นจริง	50
รูปที่ 4-4 แสดงการทำงานของคำสั่ง if เมื่อเงื่อนไขเป็นเท็จ	51
รูปที่ 4-5 แสดงการทำงานของคำสั่ง if-else	52
รูปที่ 4-6 แสดงการทำงานของคำสั่ง while	53
รูปที่ 4-7 แสดงการทำงานของคำสั่ง repeat-until	54
รูปที่ 5-1 รูปแสดงหน้าต่างหลักของโปรแกรม	59
รูปที่ 5-2 แสดงการเขียนสคริปต์เพื่อเปิดบราวเซอร์	62
รูปที่ 5-3 แสดงเว็บเพจของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	62
รูปที่ 5-4 แสดงการเขียนสคริปต์เพื่อปิดบราวเซอร์	63

รูปที่ 5-5 แสดงการเขียนสคริปต์เพื่อคำนวณอัตราแลกเปลี่ยนเงินตราและส่งเพจ	63
รูปที่ 5-6 แสดงการเขียนสคริปต์เพื่อค้นหาข้อมูลที่ต้องการ	64
รูปที่ 5-7 แสดงการเขียนสคริปต์สั่งให้บราวเซอร์เปิดและปิดตามเวลาที่กำหนด	64
รูปที่ 5-8 แสดงเงื่อนไข if และ if-else	65
รูปที่ 5-9 แสดงเงื่อนไข while และ repeat	65
รูปที่ 5-10 แสดงภาพโปรแกรมเทลเน็ต	66
รูปที่ 5-11 แสดงการระบุชื่อ โฮสต์และหมายเลขพอร์ต	66
รูปที่ 5-12 แสดงการกรอก Login Name	67
รูปที่ 5-13 แสดงการกรอก Password	67
รูปที่ 5-14 แสดงคำสั่ง runscript	68
รูปที่ 5-15 แสดงคำสั่ง editscript	68
รูปที่ 5-16 แสดงคำสั่ง exit	69
รูปที่ 5-17 แสดงหน้าต่างหลักของโปรแกรมอ่านข่าวจากนิวส์เซิร์ฟเวอร์	70
รูปที่ 5-18 แสดงฟอร์มรับ News Server และ News Group	70
รูปที่ 5-19 แสดงโปรแกรมอ่านข่าวที่ไปอ่านข่าวจาก News Server แล้ว	71
รูปที่ 5-20 แสดงฟอร์มรับข้อมูลของผู้ใช้	71

# บทที่ 1

## บทนำ

### 1.1 ที่มาของโครงการ

ในปัจจุบันนี้ บทบาทของเครือข่ายอินเทอร์เน็ตได้ทวีความสำคัญต่อการดำรงชีวิตของมนุษย์อย่างชัดเจน เนื่องจากเป็นแหล่งในการเก็บรวบรวมข้อมูลและข่าวสารต่าง ๆ มากมาย การให้บริการข้อมูลบนเครือข่ายอินเทอร์เน็ตได้มีการนำเสนอในรูปแบบต่าง ๆ กันออกไป เช่น บริการข้อมูลเพื่อการศึกษา โดยเฉพาะบริการข้อมูลเพื่อประโยชน์ในการประชาสัมพันธ์ รวมทั้งการบริการข้อมูลเพื่อประโยชน์ในเชิงธุรกิจด้วย การบริการข้อมูลจึงเป็นประโยชน์และได้รับความสนใจจากผู้ใช้ในทุกระดับชั้น จึงทำให้ขนาดของแหล่งเก็บข้อมูลใหญ่ขึ้นเพื่อการครอบคลุมการบริการ และสามารถตอบสนองความต้องการในการใช้บริการข้อมูลซึ่งแตกต่างกันได้อย่างทั่วถึง

ผู้ใช้สามารถใช้บริการเครือข่ายทางอินเทอร์เน็ตโดยผ่านเว็บเบราว์เซอร์ (Web Browser) โดยผู้ใช้สามารถกรอกยูอาร์แอล (URL) ที่ผู้ใช้ต้องการให้แก่เบราว์เซอร์ ซึ่งเบราว์เซอร์จะทำการติดต่อกับเซิร์ฟเวอร์ (Server) เพื่อค้นหาเว็บเพจ (Web Page) ที่ต้องการ และสุดท้ายเบราว์เซอร์จะแสดงเว็บเพจเหล่านั้นให้ผู้ใช้ได้เห็น

จากการทำงานของเบราว์เซอร์ในลักษณะข้างต้น จะเห็นว่าการใช้งานเบราว์เซอร์นั้นเหมาะที่จะใช้ในงานที่ผู้ใช้ต้องสามารถรับและตอบสนองบริการซึ่งแสดงออกทางเบราว์เซอร์ได้ จึงเป็นเรื่องที่ไม่สะดวกนักถ้าผู้ใช้มีเวลาน้อย แต่ต้องการเรียกใช้บริการทางเครือข่ายหลายอย่างและต้องรอรับการให้บริการเป็นเวลานาน

ทางคณะผู้จัดทำโครงการ จึงได้ริเริ่มพัฒนาเว็บเบราว์เซอร์ให้มีความสามารถมากขึ้น คือทำให้เบราว์เซอร์เป็นเบราว์เซอร์ที่สามารถถูกโปรแกรมได้โดยผู้ใช้ (Programmable Browser) ผู้ใช้สามารถตั้งงานผ่านทางเบราว์เซอร์ได้ โดยมีการนำเทคโนโลยีเอเจนต์ (Agent Technology) มาใช้เพื่ออำนวยความสะดวกให้แก่ผู้ใช้งาน โดยกำหนดให้มีโปรแกรมทำงานหรือทำหน้าที่เสมือนเป็นผู้ใช้ โดยรับคำสั่งจากผู้ใช้และทำงานตามคำสั่งที่ได้รับมอบหมาย และผลลัพธ์ที่ได้คือข้อมูลที่ตรงกับความต้องการของผู้ใช้มากที่สุด โดยผู้ใช้ไม่ต้องมาทำงานตามเวลานั้น ๆ ด้วยตัวเอง

### 1.2 วัตถุประสงค์

1. เพื่อศึกษาลักษณะการทำงานของเอเจนต์
2. เพื่อศึกษาลักษณะการทำงานของเว็บเบราว์เซอร์
3. เพื่อประยุกต์การทำงานของเอเจนต์
4. เพื่อประยุกต์การทำงานของเว็บเบราว์เซอร์
5. เพื่อพัฒนาเว็บเบราว์เซอร์ให้มีความสามารถมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 1.3 เป้าหมาย

บราวเซอร์ที่สามารถถูกโปรแกรมโดยผู้ใช้ได้

### 1.4 ขอบเขตของโครงการ

1. สร้างบราวเซอร์ต้นแบบที่สามารถทำงานหลัก ๆ ของบราวเซอร์ทั่วไปได้
2. บราวเซอร์ต้นแบบที่สร้างขึ้นสามารถรับคำสั่งจากผู้ใช้ได้
3. ผู้ใช้สามารถส่งงานบราวเซอร์ได้ด้วยภาษาสคริปต์ที่ขณะผู้จัดทำพัฒนาขึ้น
4. บราวเซอร์สามารถทำงานตามคำสั่งของผู้ใช้ได้อย่างถูกต้อง
5. บราวเซอร์สามารถพัฒนาต่อให้มีความสามารถมากขึ้นได้

### 1.5 วิธีการดำเนินงาน

การทำงานแบ่งออกเป็น 3 ส่วน ดังนี้

1. รวบรวมและศึกษาข้อมูลที่สำคัญ ดังต่อไปนี้
  - 1.1 ลักษณะทั่วไปของเเจนต์
  - 1.2 หลักการทำงานที่สำคัญของเเจนต์
  - 1.3 ข้อมูลทั่วไปของเเจนต์
  - 1.4 ข้อมูลทั่วไปของบราวเซอร์
  - 1.5 หลักการทำงานของบราวเซอร์
  - 1.6 หลักการทำงานของซีจีไอ
2. วิเคราะห์ข้อมูลที่ได้ทำการรวบรวมมาทั้งหมด เพื่อกำหนดขอบเขตและลักษณะของชิ้นงาน
3. พัฒนาตัวอย่าง โครงการขนาดเล็ก โดยมีขั้นตอน ดังนี้
  - 3.1 ออกแบบโครงการตัวอย่าง
  - 3.2 ตรวจสอบลักษณะการทำงานของโครงการตัวอย่าง ว่าสามารถทำงานได้ตามที่กำหนด
  - 3.3 ถ้าโครงการตัวอย่างทำงานไม่ตรงตามลักษณะของงานที่กำหนดไว้ ต้องทำการแก้ไขให้ให้โครงการตัวอย่างทำงานได้ถูกต้อง
  - 3.4 ถ้าโครงการตัวอย่างทำงานได้ถูกต้องตามลักษณะของงานที่กำหนดไว้ ต้องทำการพัฒนาความสามารถของโครงการของโครงการตัวอย่างให้เพิ่มขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### หลักการพื้นฐาน

#### 2.1 หลักการพื้นฐานของเอเจนต์ (Agents)

##### 2.1.1 ความหมายของเอเจนต์

เอเจนต์ในมุมมองของผู้ใช้ คือ โปรแกรมที่ช่วยงานผู้ใช้และทำงานที่เป็นประโยชน์กับผู้ใช้

เอเจนต์ในมุมมองของระบบ คือ โปรแกรมที่

- อาศัยอยู่ในเอ็กซิคิวต์ชันเอนไวรอนเมนต์ (execution environment)
- มีคุณสมบัติดังต่อไปนี้
  - รีแอ็กทีฟ (Reactive) รับรู้การเปลี่ยนแปลงของสิ่งแวดล้อมและตอบสนองต่อการเปลี่ยนแปลงของสิ่งแวดล้อม
  - ออโตโนมัส (Autonomous) ควบคุมการทำงานของตัวเองได้สามารถทำกิจกรรมต่างๆ ได้อย่างอิสระ โดยไม่ขึ้นกับผู้ใช้ หรือสิ่งกระตุ้นภายนอก
  - มีเป้าหมายแน่นอน (Goal-driven)
  - ทำงานอย่างต่อเนื่อง (Continuous)
- และอาจจะมีคุณสมบัติบางอย่างดังนี้
  - สามารถติดต่อกับเอเจนต์ตัวอื่นได้ (Communicative)
  - สามารถเคลื่อนย้ายจากที่หนึ่งไปยังอีกที่หนึ่งได้ แม้ว่าจะอยู่คนละแพลตฟอร์มกัน (Mobile)
  - สามารถเรียนรู้และเปลี่ยนแปลงพฤติกรรมได้ขึ้นอยู่กับประสบการณ์ (Adaptive)

##### 2.1.2 การแบ่งประเภทของเอเจนต์

เอเจนต์อาจจะแบ่งประเภทได้ตามคุณสมบัติดังกล่าวนี้ หรือจะแบ่งประเภทของเอเจนต์ด้วยวิธีอื่นก็ได้ เช่น เราอาจจะแบ่งเอเจนต์ตามงานที่เอเจนต์ทำ ตัวอย่างเช่น เอเจนต์ที่ทำหน้าที่รวบรวมข้อมูลหรือเอเจนต์ที่ทำหน้าที่จองรถจักรยานยนต์

Brustoloni แบ่งเอเจนต์ออกเป็น 3 ประเภท คือ เรกกูเลชันเอเจนต์ (regulation agents), แพลนนิ่งเอเจนต์ (planning agents) และออดาปทีฟเอเจนต์ (adaptive agents) เรกกูเลชันเอเจนต์ คือ เอเจนต์ที่ตอบสนองต่อการเปลี่ยนแปลงของสภาพแวดล้อม และเป็นเอเจนต์ที่มีเป้าหมายว่าจะทำอะไรอย่างแน่นอน แพลนนิ่งเอเจนต์ เป็นเอเจนต์ที่เลือกวิธีการทำงานที่เหมาะสมที่สุด โดยใช้ทฤษฎีของวิชาปัญญาประดิษฐ์ หรืออาจจะใช้อัลกอริ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทีมแบบสุ่ม (Randomizing Agent) อะแคปทีฟเอเจนต์ เป็นเอเจนต์ที่สามารถเรียนรู้และเปลี่ยนแปลงการทำงานของตัวเองได้ หรืออาจจะเรียกว่า เอเจนต์ที่สามารถเรียนรู้ได้ (Learning Agent)

Property	Other Names	Meaning
Reactive	(sensing and acting)	ตอบสนองต่อการเปลี่ยนแปลงของสิ่งแวดล้อม
Autonomous		สามารถทำงานได้โดยอัตโนมัติ
Goal-oriented	pro-active purposeful	มีเป้าหมายในการทำงาน
Temporally continuous		ทำงานอย่างต่อเนื่อง
Communicative	Socially able	ติดต่อกับเอเจนต์อื่นๆ หรือผู้ใช้
Learning	Adaptive	เรียนรู้และเปลี่ยนแปลงการทำงานของตัวเองตามประสบการณ์
Mobile		สามารถเคลื่อนย้ายตัวเองจากที่หนึ่งไปอีกที่หนึ่ง
Flexible		มีความยืดหยุ่น

### ตารางที่ 2-1 แสดงคุณสมบัติต่างๆ ของเอเจนต์

#### 2.1.3 ลำดับชั้นของเอเจนต์

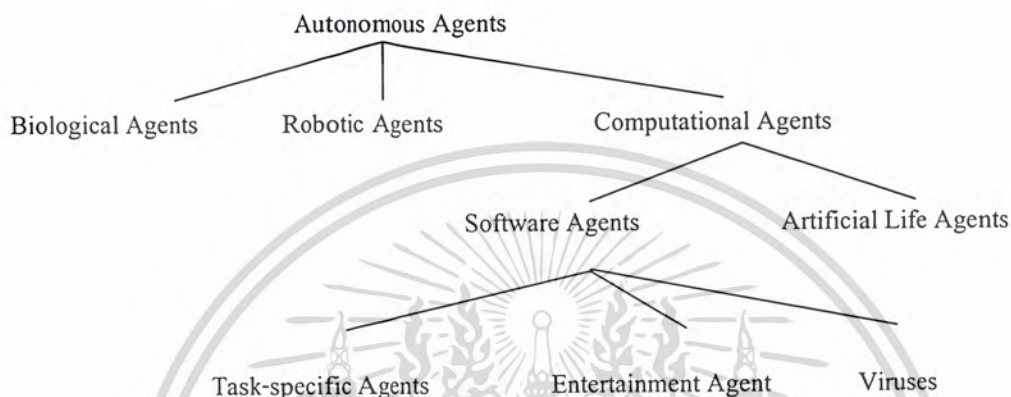
เราสามารถเปรียบเทียบความสัมพันธ์ระหว่างเอเจนต์ชนิดต่าง ๆ กับ โมเดลเชิงชีววิทยาได้ ตัวอย่างเช่น ลำดับชั้นของมนุษย์เป็นดังนี้

- Kingdom = Animal
- Phylum = Chordata
- Class = Mammalia
- Order = Primate
- Family = Pongidae
- Subfamily = Hominidae
- Genus = Homo
- Species = Sapiens

เราสามารถจัดลำดับชั้นของเอเจนต์ให้อยู่ในลักษณะนี้ได้เช่นกัน โดยในระดับ Kingdom เราสามารถแบ่งเอเจนต์ออกเป็น 3 ชนิดดังนี้ ไบโอบีโอดีคอลลเอเจนต์ (Biological agent), โรโบติกคอลลเอเจนต์ (Robotic agent) และคอมพิวเตอร์เอเจนต์ (Computational agent) โดยเอเจนต์ที่เราสนใจอยู่ในกลุ่มของคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชั้นนอลเอเจนต์ ในระดับ Phylum เราแบ่งคอมพิวเตอร์ชั้นนอลเอเจนต์ได้เป็น 2 ประเภท คือ ซอฟต์แวร์เอเจนต์ (Software Agents) และ อาชีพิชีวลไลฟ์เอเจนต์ (Artificial life agents) เราสามารถแบ่งซอฟต์แวร์เอเจนต์ได้เป็น 3 ประเภท ทาส์กสเปคิฟิเคเอเจนต์ (Task-specific agents), เอนเตอร์เทนเมนต์เอเจนต์ (entertainment agents) และไวรัสคอมพิวเตอร์ (computer viruses) เราสามารถแสดงเป็นภาพได้ดังนี้



รูปที่ 2-1 แสดงลำดับชั้นของเอเจนต์

#### 2.1.4 การใช้งานสำหรับเอเจนต์ที่ชาญฉลาด

คุณสมบัติต่าง ๆ ของภาษาจาวาที่สนับสนุนอินเทลลิเจนต์เอเจนต์แอปพลิเคชัน (Intelligent Agent Application) มีดังนี้ ความเป็นอัตโนมัติ (Autonomous), ความฉลาด (Intelligence) และความสามารถเคลื่อนที่ได้ (Mobility)

##### ความเป็นอัตโนมัติ

โปรแกรมที่สามารถทำงานโดยอัตโนมัติได้ คือ โปรแกรมที่มีการแบ่งออกเป็นโพรเซสต่าง ๆ โปรแกรมที่สร้างด้วยภาษาจาวาก็สามารถแบ่งการทำงานออกเป็นหลาย ๆ โพรเซสได้ และแต่ละโพรเซสสามารถทำงานได้โดยอิสระ โปรแกรมที่เขียนด้วยภาษาจาวาสามารถเชื่อมต่อกับโปรแกรมอื่น ๆ ได้โดยอาศัยซ็อกเก็ต (sockets) ภาษาจาวาสนับสนุน โปรแกรมที่มีการทำงานเป็นเธรด (threaded)

โปรแกรมที่มีลักษณะการทำงานเป็นอัตโนมัติจะทำงานอยู่ตลอดเวลา พร้อมทั้งจะตอบสนองต่อผู้ใช้หรือการกระตุ้นจากสิ่งแวดล้อมได้ทุกเมื่อ คำถามที่เกิดขึ้นคือ เอเจนต์รู้ได้อย่างไรว่าเมื่อไรมีการเปลี่ยนแปลงในสิ่งแวดล้อม ในโมเดลที่เราใช้ เอเจนต์จะรู้ว่ามีเปลี่ยนแปลงเมื่อมีการส่งอีเวนต์ (Event) มาถึงเอเจนต์จากมุมมองการออกแบบเชิงวัตถุ อีเวนต์ คือ การเรียกเมธอด (Method) หรือการส่งเมสเสจ (Message) นั่นเอง

ในภาษาจาวามีกลไกที่จัดการกับอีเวนต์ที่ผู้ใช้สร้างขึ้น เช่น การเลื่อนเมาส์ และการเลือกเมนู เป็นต้น

## ความฉลาด

ในการสร้างโปรแกรมปัญญาประดิษฐ์ (AI Applications) มีปัจจัยหลัก ๆ 2 อย่าง คือ knowledge representation และอัลกอริทึมที่ใช้ ภาษายาวาให้การสนับสนุนมาตรฐาน AI Knowledge representation เช่น if-then rule

## ความสามารถเคลื่อนที่ได้

ไบนารีโค้ด (Bytes code) ของภาษายาวา และไฟล์นามสกุล JAR สามารถทำให้คลาสต่าง ๆ ที่คอมไพล์แล้ว สามารถส่งผ่านเครือข่ายได้ และถูกเอ็กซีกิวต์ที่เครื่องเป้าหมาย (target machine)

หนึ่งในความต้องการหลัก ๆ ของโปรแกรมที่มีความสามารถในการเคลื่อนที่ได้คือ ความสามารถในการเก็บสถานะเอาไว้ได้ และสามารถนำกลับมาทำงานต่อจากสถานะเดิมได้

## 2.2 หลักการพื้นฐานของเว็บเบราว์เซอร์

### 2.2.1 โปรแกรมเว็บเบราว์เซอร์

คือ โปรแกรมที่ใช้แสดงเว็บเพจต่างๆ เราสามารถแบ่งเบราว์เซอร์ตามลักษณะการแสดงผลได้ 2 ประเภท ดังนี้

1. เท็กซ์เบราว์เซอร์ (Text Browser) เป็นเบราว์เซอร์ที่มีมาก่อน แต่ในปัจจุบันใช้ในการศึกษาและวิจัย ไม่เป็นที่นิยมเพราะแสดงผลเป็นตัวอักษรอย่างเดียว
2. กราฟิกคอลเลอร์เบราว์เซอร์ (Graphical Browser) เป็นเบราว์เซอร์ที่มากถูกพัฒนาขึ้นมาทีหลัง เบราว์เซอร์ชนิดนี้สามารถแสดงรูปภาพต่าง ๆ ได้ ในเบราว์เซอร์ที่มีความทันสมัยมาก ๆ สามารถแสดงผลได้ในรูปสื่อผสม (Multimedias) คือมีทั้งภาพและเสียง ในอนาคตคาดว่าเบราว์เซอร์จะสามารถแสดงผลในรูปของกลิ่นได้

ตัวอย่างเบราว์เซอร์และรายละเอียด

#### 1. โมเสก (Mosaic)

โมเสกเป็นกราฟิกคอลเลอร์เบราว์เซอร์ตัวแรกที่มีการแสดงในลักษณะของรูปภาพสร้างในปี 1993 โดย Mark Andreessen เป็นผู้ริเริ่มการสร้างเบราว์เซอร์ตัวนี้ ร่วมกับ National Center for Supercomputing Application (NCSA) ใช้ได้บนระบบปฏิบัติการวินโดวส์ 3.1X (Windows v.3.1.x), วินโดวส์ 95 (Windows 95) และ วินโดวส์ เอ็นที (Windows NT)

#### 2. เน็ตสเคป (Netscape)

เน็ตสเคปเป็นเว็บเบราว์เซอร์ที่แสดงด้วยรูปภาพและเป็นที่นิยมของผู้ใช้ทั่วไป พัฒนาโดยบริษัทเน็ตสเคป คอมมิวนิเคชัน (Netscape Communications) เบราว์เซอร์ของเน็ตสเคปเรียกว่า นาวิกเตอร์ (Navigator) ซึ่งถูกพัฒนาในปี 1995 โดยมี Mark Andreessen เป็นหัวหน้าในการพัฒนา (ซึ่งเป็นผู้สร้างโมเสก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3. ไมโครซอฟท์ อินเทอร์เน็ต เอ็กซ์พลอเรอร์ (Microsoft Internet Explorer)

ไมโครซอฟท์ อินเทอร์เน็ต เอ็กซ์พลอเรอร์ เป็นบราวเซอร์ที่แสดงรูปภาพด้วยเช่นกัน ใช้บนระบบปฏิบัติการ วินโดวส์ 95, วินโดวส์ 98 และ วินโดวส์ เอ็นที

### 4. ลินซ์ (Lynx)

ลินซ์ เป็นบราวเซอร์ที่แสดงด้วยตัวอักษร (Text Browser) ซึ่งพัฒนาโดยมหาวิทยาลัยแคนซัส (University of Kansas) โดยใช้บนยูนิกซ์เวิร์กสเตชัน (UNIX workstation)

### 5. โอเปรา (Opera)

โอเปราเป็นบราวเซอร์ที่มีชื่อดีกว่าบราวเซอร์ของเน็ตสเคป และ ไมโครซอฟท์ อินเทอร์เน็ต เอ็กซ์พลอเรอร์ (ซึ่งเป็นบราวเซอร์ที่ได้รับความนิยมมาก) คือ เป็นบราวเซอร์ที่มีขนาดเล็ก ใช้เวลาในการโหลดน้อย โอเปราถูกพัฒนาโดย บริษัทนอร์วีเจียน ซอฟท์แวร์ (Norwegian software company) ซึ่งสามารถใช้ได้บนระบบปฏิบัติการวินโดวส์ และ โอเอสทู (OS/2)

### 6. ฮอตจาวา (HotJava)

ฮอตจาวา เป็นบราวเซอร์ที่แสดงด้วยรูปภาพ โดยมี บริษัทซันไมโครซิสเต็มส์ (Sun Microsystems) เป็นผู้พัฒนา ในเวอร์ชันใหม่ล่าสุดมีการใช้คอมโพเนนต์ (Component) ของจาวาบีนส์ (JavaBean) ด้วย ลักษณะเด่นของฮอตจาวา บราวเซอร์ (HotJava Browser) คือ ความสามารถในการใช้จาวาสคริปต์ (JavaScript)

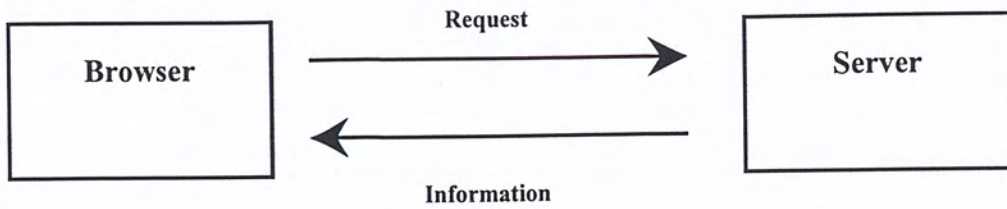
รายชื่อ	ผู้พัฒนา	Operating system
Aweb II	Amitrix	Amiga
Hot Java	Sun Microsystem	Mac, OS/2, Windows 98
Lynx	Lynx team developer	Amiga, Mac, OS/2
Netscape	Netscape	Windows XX
Internet Explorer	Microsoft	Windows XX

ตารางที่ 2-2 ตัวอย่างบราวเซอร์ที่มีในปัจจุบัน

#### 2.2.2 หลักการทำงานของโปรแกรมเว็บเบราว์เซอร์

บราวเซอร์หรือเราอาจจะเรียกว่าเป็นเอเจนต์ของผู้ใช้ ทำงานโดยใช้โพรโทคอลที่เรียกว่า ไฮเพอร์เท็กซ์ทรานสปอร์ตโพรโทคอล (HyperText Transport Protocol หรือ HTTP) เพื่อขอข้อมูลจากเว็บเซิร์ฟเวอร์ (web server) ข้อมูลนี้เขียนขึ้นด้วยภาษาไฮเพอร์เท็กซ์มาร์คอัพ (HyperText Markup Language หรือ HTML) ไฟล์ชนิดนี้จะถูกอินเทอร์พรีต (interpreted) โดยเว็บเบราว์เซอร์ เอกสารเอชทีเอ็มแอล สามารถอ้างอิงไปยังเว็บไซต์อื่น ๆ ได้โดยใช้ไฮเพอร์ลิงก์ (hyperlink)

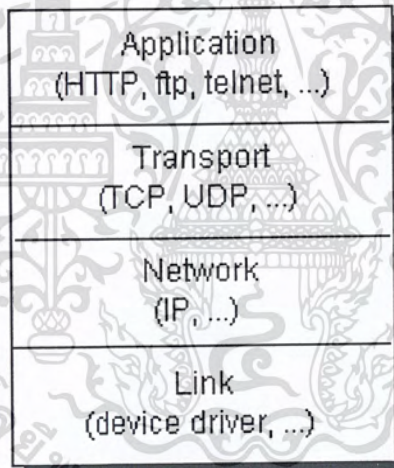
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-2 แสดงลักษณะการทำงานของบราวเซอร์

### 2.3 พื้นฐานต่าง ๆ เกี่ยวกับเครือข่าย ( Networking Basics )

คอมพิวเตอร์ต่าง ๆ บนเครือข่ายอินเทอร์เน็ตติดต่อสื่อสารระหว่างกันผ่านทรานสมิซชันคอนโทรล โพรโตคอล (Transmission Control Protocol) หรือ ยูสเซอร์ดาต้าแกรม โพรโตคอล (User Datagram Protocol) ดังภาพต่อไปนี้



รูปที่ 2-3 แสดงเลเยอร์ต่างๆ

ในขณะที่ผู้เขียนโปรแกรมเขียนโปรแกรมภาษาจาวาที่มีการติดต่อสื่อสารผ่านเครือข่าย นั่นคือผู้เขียนโปรแกรมกำลังเขียนโปรแกรมบน Application Layer โดยปกติแล้วผู้เขียนโปรแกรมไม่จำเป็นต้องเกี่ยวข้องกับ ทรานสมิซชันคอนโทรล โพรโตคอล และ ยูสเซอร์ดาต้าแกรม โพรโตคอล โดยตรง แต่ผู้เขียนโปรแกรมสามารถใช้ คลาสต่าง ๆ ในแพ็คเกจ java.net คลาสต่าง ๆ เหล่านี้จะให้การติดต่อสื่อสารผ่านเครือข่ายโดยไม่มีสนใจระบบ ( system-independent ) อย่างไรก็ตามในการเลือกที่จะใช้ คลาสใดในโปรแกรม ผู้เขียนโปรแกรมจะต้องรู้ว่า ทรานสมิซชันคอนโทรล โพรโตคอล กับ ยูสเซอร์ดาต้าแกรม โพรโตคอล แตกต่างกันอย่างไรก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ทรานส์มิสชันคอนโทรลโพรโตคอล

ทรานส์มิสชันคอนโทรลโพรโตคอลเป็นโพรโตคอลที่มีการสร้างการเชื่อมต่อขึ้นก่อนมีการติดต่อสื่อสาร (connection-based protocol) เมื่อโปรแกรมสองโปรแกรมต้องการจะติดต่อสื่อสารระหว่างกัน โปรแกรมทั้งสองจะต้องสร้างการเชื่อมต่อ (connection) และส่งข้อมูลไป-มาระหว่างโปรแกรมโดยผ่านการเชื่อมต่อที่สร้างขึ้นมา การติดต่อในลักษณะนี้คล้ายกับการโทรศัพท์ เช่น ถ้าผู้เขียนโปรแกรมต้องการพูดคุยกับเพื่อนที่อยู่กันคนละจังหวัด การเชื่อมต่อจะถูกสร้างขึ้นมาเมื่อผู้เขียนโปรแกรมต่อโทรศัพท์ไปหาเพื่อนแล้วเพื่อนรับสาย ผู้เขียนโปรแกรมส่งข้อมูลไป-มาผ่านการเชื่อมต่อที่สร้างขึ้นมาโดยการพูดผ่านสายโทรศัพท์ไปยังอีกฝ่าย ทรานส์มิสชันคอนโทรลโพรโตคอล รับประกันว่าข้อมูลถูกส่งผ่านการเชื่อมต่อจากปลายทางด้านหนึ่งไปยังปลายทางอีกด้านหนึ่ง

ทรานส์มิสชันคอนโทรลโพรโตคอล จัดให้มีช่องทางการเชื่อมต่อแบบหนึ่ง-หนึ่ง (point-to-point channel) สำหรับแอปพลิเคชันต่าง ๆ ที่ต้องการการติดต่อสื่อสารที่น่าเชื่อถือ เชททีพี (HTTP), เอฟทีพี (FTP) และ เทลเน็ต (Telnet) คือตัวอย่างของแอปพลิเคชันที่ต้องการการติดต่อสื่อสารที่น่าเชื่อถือ เพราะว่าลำดับของข้อมูลที่ส่งและรับผ่านเครือข่ายมีความจำเป็นมากสำหรับแอปพลิเคชันเหล่านี้ เช่น เชททีพีถูกใช้เพื่ออ่านข้อมูลจากยูอาร์แอล (URL) ข้อมูลที่รับมาต้องอยู่ในลำดับที่ถูกต้องตามลำดับที่ถูกส่งมา

## ยูสเซอร์ดาต้าแกรมโพรโตคอล

ยูสเซอร์ดาต้าแกรมโพรโตคอลให้การติดต่อสื่อสารระหว่างเครือข่ายโดยไม่มีการรับประกันว่าข้อมูลที่ถูกส่งระหว่างแอปพลิเคชันจะไปถึงปลายทาง ยูสเซอร์ดาต้าแกรมโพรโตคอลไม่เหมือนทรานส์มิสชันคอนโทรลโพรโตคอลที่มีการสร้างการเชื่อมต่อขึ้นก่อนการติดต่อ แต่ยูสเซอร์ดาต้าแกรมโพรโตคอลจะส่งแพ็กเก็ตต่าง ๆ ของข้อมูลอย่างอิสระ จากแอปพลิเคชันหนึ่งไปยังอีกแอปพลิเคชัน แพ็กเก็ตต่าง ๆ ของข้อมูลเหล่านี้เรียกว่า ดาต้าแกรม (datagrams) การส่งดาต้าแกรมเหมือนกับการส่งจดหมายผ่านบริการไปรษณีย์ ลำดับของการนำส่งไม่มีนัยสำคัญและไม่มีการรับรองว่าจดหมายจะถูกส่งถึงมือผู้รับ

## พอร์ต (port)

โดยทั่ว ๆ ไปคอมพิวเตอร์เครื่องหนึ่งจะมีการเชื่อมต่อทางกายภาพไปยังเครือข่ายเพียงการเชื่อมต่อเดียว ทุก ๆ ข้อมูลที่ถูกส่งออกไปจะมุ่งไปยังคอมพิวเตอร์ที่อยู่ปลายทางการเชื่อมต่อ แต่อย่างไรก็ตาม ข้อมูลอาจจะถูกส่งไปยังคนละแอปพลิเคชันที่ทำงานอยู่บนเครื่องคอมพิวเตอร์ปลายทาง แล้วเครือข่ายจะรู้ได้อย่างไรว่าข้อมูลใดเป็นของแอปพลิเคชันใด ในการใช้พอร์ตข้อมูลที่ถูกส่งผ่านเครือข่ายอินเทอร์เน็ตจะถูกส่งไปพร้อมกับข้อมูลระบุที่อยู่ (addressing information) ซึ่งจะบอกคอมพิวเตอร์และพอร์ตปลายทาง คอมพิวเตอร์ปลายทางจะถูกระบุด้วย ไอพีแอดเดรส (IP address) ขนาด 32 บิต ซึ่งเป็นข้อมูลที่ อินเทอร์เน็ตโพรโตคอล ใช้ นำส่ง

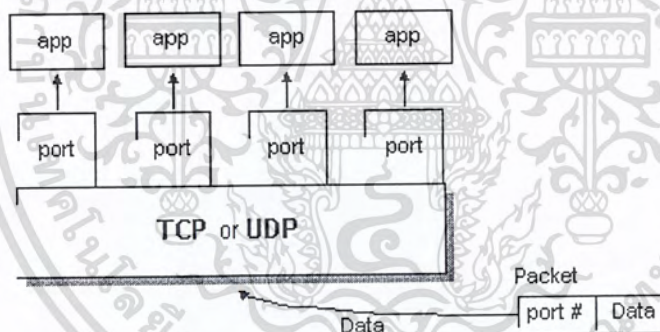
ข้อมูลไปยังเครื่องปลายทางที่ต้องการ พอร์ตต่าง ๆ ระบุด้วย เลขขนาด 16 บิต ซึ่งเป็นตัวเลขที่ ทรานส์มิสชันคอนโทรลโพรโทคอล และ ยูสเซอร์ค่าคำแกรมโพรโทคอลให้นำส่งข้อมูลไปยังแอปพลิเคชันที่ต้องการ

ในโพรโทคอลที่ต้องสร้างการเชื่อมต่อขึ้นมาก่อนการติดต่อสื่อสาร เช่น ทรานส์มิสชันคอนโทรลโพรโทคอล เซิร์ฟเวอร์จะเชื่อมต่อซ็อกเก็ตเข้ากับหมายเลขพอร์ตที่ระบุเอาไว้ ข้อมูลต่าง ๆ ที่ถูกส่งผ่านเครือข่ายจะเข้ามาทางพอร์ตนี้ โคลเอ็นต์สามารถติดต่อกับเซิร์ฟเวอร์ได้ทางพอร์ตนี้ ดังภาพต่อไปนี้



รูปที่ 2-4 แสดงการติดต่อระหว่างโคลเอ็นต์และเซิร์ฟเวอร์

ในโพรโทคอลที่ไม่ต้องสร้างการเชื่อมต่อขึ้นมาก่อน เช่น ยูสเซอร์ค่าคำแกรมโพรโทคอล คำคำแกรมแพ็กเก็ตจะประกอบด้วยหมายเลขพอร์ตปลายทาง และยูสเซอร์ค่าคำแกรมโพรโทคอลจะส่งแพ็กเก็ตไปยังแอปพลิเคชันที่เหมาะสม ดังภาพต่อไปนี้



รูปที่ 2-5 แสดงการติดต่อโดยใช้โพรโทคอล

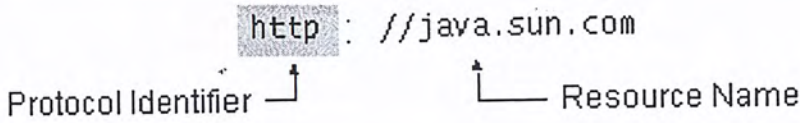
หมายเลขพอร์ตอยู่ในช่วง 0-65,535 เพราะว่าหมายเลขพอร์ตระบุด้วยเลขขนาด 16 บิต หมายเลขพอร์ตที่อยู่ในช่วง 0-1023 ถูกสงวนไว้สำหรับบริการมาตรฐาน เช่น เอชทีทีพี และ เอพทีพี เป็นต้น หมายเลขพอร์ตเหล่านี้เป็นหมายเลขพอร์ตที่เราไม่ควรนำมาใช้

### 2.3.1 ยูอาร์แอล (URL)

ยูอาร์แอล (URL: Uniform Resource Locator) คือ ตัวอ้างอิง (reference) ไปยังรีซอร์ส (resource) บนอินเทอร์เน็ต ยูอาร์แอลเป็นรูปแบบของข้อความที่อธิบายว่าจะพบรีซอร์สบนอินเทอร์เน็ตได้อย่างไร ยูอาร์แอลประกอบด้วยสองส่วนหลัก ๆ คือ โพรโทคอลที่ใช้ และ ที่อยู่ของรีซอร์ส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างต่อไปนี้เป็นตัวอย่างของยูอาร์แอลที่อยู่ของเว็บไซค์จาวา



รูปที่ 2-6 แสดงตัวอย่างของยูอาร์แอลที่อยู่ของเว็บไซค์จาวา

จากภาพนี้จะเห็นได้ว่ายูอาร์แอลแบ่งออกเป็นสองส่วนหลัก ๆ คือ

- ส่วนที่ระบุโปรโตคอล
- ส่วนที่ระบุชื่อรีซอร์ส

สังเกตว่าทั้งสองส่วนแยกกันด้วยเครื่องหมาย “://” ส่วนที่ระบุโปรโตคอลจะบอกโปรโตคอลที่ใช้เพื่อไปดึงเอาข้อมูลมา ในตัวอย่างนี้ใช้โปรโตคอลเอชทีทีพี ซึ่งเอชทีทีพีเป็นหนึ่งในหลาย ๆ โปรโตคอลที่ใช้เข้าถึงรีซอร์สชนิดต่าง ๆ บนอินเทอร์เน็ต โปรโตคอลอื่น ๆ ที่ใช้ได้ เช่น ไฟล์ทรานส์เฟอร์โปรโตคอล (FTP), โกเฟอร์ (gopher), ไฟล์ (File) และ นิวส์ (News)

รูปแบบของส่วนที่ระบุชื่อรีซอร์ส ขึ้นอยู่กับโปรโตคอลที่ใช้ สำหรับโปรโตคอลส่วนใหญ่ รวมทั้งเอชทีทีพี ส่วนที่ระบุชื่อรีซอร์สจะประกอบด้วยองค์ประกอบต่าง ๆ ที่อยู่ในตารางที่ 2-3 อย่างน้อยหนึ่งอย่าง

ตารางที่ 2-3 แสดงองค์ประกอบต่าง ๆ ของส่วนที่ระบุชื่อรีซอร์ส

ชื่อโฮสต์	ชื่อของเครื่องที่เก็บรีซอร์ส
ชื่อไฟล์	พาทของไฟล์ที่อยู่บนเครื่อง
หมายเลขพอร์ต	หมายเลขพอร์ต (มีหรือไม่มีก็ได้)

สำหรับโปรโตคอลทั่ว ๆ ไป ต้องระบุชื่อโฮสต์และชื่อไฟล์ โดยที่หมายเลขพอร์ตจะมีหรือไม่มีก็ได้ ตัวอย่างเช่น ส่วนที่ระบุชื่อรีซอร์สสำหรับโปรโตคอลเอชทีทีพี ต้องระบุเซิร์ฟเวอร์บนเครือข่าย หรือ ชื่อโฮสต์ และ พาทของเอกสารที่อยู่บนเครื่อง หรือชื่อไฟล์ ในที่นี้อาจจะระบุพอร์ตด้วยก็ได้

### การเชื่อมต่อไปยังยูอาร์แอล

ในภาษาจาวาถ้าผู้เขียนโปรแกรมต้องการมากกว่าแค่การอ่านจากยูอาร์แอล ผู้เขียนโปรแกรมต้องติดต่อไปยังยูอาร์แอลโดยการเรียก openConnection() บนยูอาร์แอล เมธอด openConnection() จะส่ง วัตถุยูอาร์

แอลคอนเนกชัน (URLConnection Object) กลับออกมา ซึ่งวัตถุนี้สามารถใช้สำหรับการติดต่อสื่อสารได้หลายอย่าง เช่น การอ่านจากยูอาร์แอล, การเขียนไปยังยูอาร์แอล หรือการทำการควิรีไปยังยูอาร์แอลเพื่อข้อมูลต่าง ๆ

หลังจากที่สร้างวัตถุยูอาร์แอลแล้ว ผู้เขียนโปรแกรมสามารถเรียกเมธอด `openConnection()` ของวัตถุยูอาร์แอลเพื่อเชื่อมต่อไปยังยูอาร์แอลได้ ตัวอย่างเช่น ถ้าต้องการสร้างการเชื่อมต่อไปยังไซต์ Yahoo สามารถเขียนโปรแกรมได้ดังนี้

```
try {
    URL yahoo = new URL("http://www.yahoo.com/");
    URLConnection yahooConnection = yahoo.openConnection();
} catch (MalformedURLException e) { // new URL() failed
    ...
} catch (IOException e) { // openConnection() failed
    ...
}
```

เมธอด `openConnection()` จะสร้าง `URLConnection` ขึ้นมาใหม่แล้วเชื่อมต่อไปยังยูอาร์แอลนั้นแล้วคืนวัตถุ `URLConnection` กลับมา ถ้ามีความผิดพลาดเกิดขึ้น เช่น เซิร์ฟเวอร์ล้ม เมธอด `openConnection()` จะคืนค่า `IOException` กลับออกมา

เมื่อการเชื่อมต่อไปยังยูอาร์แอลประสบความสำเร็จแล้ว ผู้เขียนโปรแกรมสามารถใช้วัตถุ `URLConnection` เพื่อ อ่านจากหรือเขียนไปยังการเชื่อมต่อได้

### การอ่านจากยูอาร์แอลและการเขียนไปยังยูอาร์แอล

เมื่อการเชื่อมต่อไปยังยูอาร์แอลประสบความสำเร็จแล้วผู้เขียนโปรแกรมจะมีตัวอ้างอิงไปยังวัตถุ `URLConnection` คลาส `URLConnection` ประกอบด้วยหลาย ๆ เมธอดที่ช่วยให้ผู้เขียนโปรแกรมติดต่อสื่อสารกับยูอาร์แอลผ่านเครือข่ายได้ เมธอดต่าง ๆ ของคลาส `URLConnection` มีประโยชน์เมื่อใช้กับยูอาร์แอลของเอชทีทีพี

### การอ่านจากยูอาร์แอล

การอ่านจากยูอาร์แอลสามารถทำได้ดังนี้

```
URL yahoo = new URL("http://www.yahoo.com/");
```

```
URLConnection yc = yahoo.openConnection();
```

สร้างการเชื่อมต่อไปยังยูอาร์แอลที่ต้องการ "yc" จะเป็นตัวอ้างอิงไปยังไซต์ Yahoo

```
BufferedReader in = new BufferedReader( new InputStreamReader( yc.getInputStream()));
```

ใช้ BufferedReader ครอบคลุมไปเพื่อความสะดวกในการจัดการข้อมูล

```
String inputLine;
```

```
while ((inputLine = in.readLine()) != null)
```

```
    System.out.println(inputLine);
```

```
in.close();
```

อ่านข้อมูลเข้ามาแล้วพิมพ์ออกทางเอาต์พุตมาตรฐาน เมื่ออ่านข้อมูลหมดก็ปิดการเชื่อมต่อเพื่อคืนทรัพยากรให้กับระบบ

### การเขียนไปยังยูอาร์แอล

เพจของเอชทีเอ็มแอลจำนวนมากประกอบด้วยฟอร์ม (forms) ฟอร์ม คือ ช่องกรอกข้อมูลที่ให้ใส่ข้อมูลเพื่อส่งไปประมวลผลที่เซิร์ฟเวอร์ หลังจากที่เรากรอกข้อมูลลงในฟอร์มแล้วกดปุ่มตกลง โปรแกรมเว็บเบราว์เซอร์ของผู้ใช้จะเขียนข้อมูลไปยังยูอาร์แอลผ่านเครือข่าย ที่อีกปลายข้างหนึ่งของการเชื่อมต่อ ซีจีไอสคริปต์จะคอยทำหน้าที่ประมวลผลข้อมูลที่ส่งเข้ามาและส่งผลลัพธ์กลับไป ซึ่งโดยส่วนมากแล้วผลลัพธ์ที่ส่งกลับไปจะอยู่ในรูปของไฟล์เอชทีเอ็มแอล

โปรแกรมที่เขียนด้วยภาษาจาวาสามารถติดต่อกับซีจีไอสคริปต์ที่อยู่ฝั่งเซิร์ฟเวอร์ได้ การที่จะทำแบบนี้ได้จะต้องสามารถเขียนไปยังยูอาร์แอลได้ ซึ่งสามารถทำได้ดังนี้

1. สร้างยูอาร์แอล
2. เปิดการเชื่อมต่อไปยังยูอาร์แอล
3. สร้างความสามารถในการเขียนไปยังยูอาร์แอล
4. รับเอาต์พุตสตรีมจากการเชื่อมต่อ เอาต์พุตสตรีมนี้เชื่อมต่อเข้ากับอินพุตสตรีมมาตรฐานของซีจีไอสคริปต์บนฝั่งเซิร์ฟเวอร์
5. เขียนไปยังเอาต์พุตสตรีม
6. ปิดเอาต์พุตสตรีม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อไปนี่คือซีจีไอสคริปต์ที่ฝั่งเซิร์ฟเวอร์

```
#!/opt/internet/bin/perl
read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
@pairs = split(/&/, $buffer);
foreach $pair (@pairs) {
    ($name, $value) = split(/=/, $pair);
    $value =~ tr/+//;
    $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
    # Stop people from using subshells to execute commands
    $value =~ s/~!/~!/g;
    $FORM{$name} = $value;
}
print "Content-type: text/plain\n\n";
print "$FORM{'string'} reversed is: ";
$foo=reverse($FORM{'string'});
print "$foo\n";
exit 0;
```

สคริปต์นี้จะอ่านสตริงจากอินพุตมาตรฐาน นำมาเรียงกลับกันแล้วเขียนผลลัพธ์กลับมายังเอาต์พุตมาตรฐานของมัน

ตัวอย่าง โปรแกรมภาษาจาวาที่ติดต่อกับซีจีไอสคริปต์ข้างต้นนี้มีดังนี้

```
if (args.length != 1) {
    System.err.println("Usage: java Reverse " +
        "string_to_reverse");
    System.exit(-1);
}
String stringToReverse = URLEncoder.encode(args[0]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมนี้รับประกันว่าผู้ใช้ใส่อาร์กิวเมนต์เพียงตัวเดียวให้กับโปรแกรมผ่าน command-line แล้วทำการเข้ารหัสอาร์กิวเมนต์นั้น อาร์กิวเมนต์ตัวนี้คือสตริงที่จะถูกนำไปเรียงย้อนกลับโดยซีจีไอสคริปต์ สตริงนี้อาจจะประกอบด้วยช่องว่างและเครื่องหมายต่าง ๆ ซึ่งช่องว่างและเครื่องหมายต่าง ๆ เหล่านี้จะถูกเข้ารหัสด้วยคลาส URLEncoder โดยใช้ เมธอด encode()

```
URL url = new URL("http://java.sun.com/cgi-bin/backwards");
URLConnection c = url.openConnection();
c.setDoOutput(true);
```

แล้วสร้างวัตถุของคลาส URL ที่ชี้ไปยังซีจีไอสคริปต์ที่อยู่บนเซิร์ฟเวอร์ แล้วสร้างการเชื่อมต่อไปยังยูอาร์แอล แล้วกำหนดค่าให้สามารถเขียนเอาต์พุตออกไปยังยูอาร์แอลได้

```
PrintWriter out = new PrintWriter(c.getOutputStream());
```

แล้วสร้างเอาต์พุตสตรีมไปยังการเชื่อมต่อ และ เปิดสตรีม PrintWriter ถ้ายูอาร์แอลที่กำหนดไว้ไม่สามารถรับเอาต์พุตจากโปรแกรมได้ เมธอด getOutputStream() จะส่ง UnknownServiceException. ออกมา แต่ถ้ายูอาร์แอลที่กำหนดไว้รองรับการเอาต์พุต แล้วเมธอดนี้จะส่ง เอาต์พุตสตรีมที่เชื่อมต่อไปยังอินพุตสตรีมมาตรฐานของยูอาร์แอลบนฝั่งเซิร์ฟเวอร์ นั่นคือเอาต์พุตจากฝั่งไคลเอ็นต์ คือ อินพุตของฝั่งเซิร์ฟเวอร์

```
out.println("string=" + stringToReverse);
out.close();
```

ต่อไป โปรแกรมจะเขียนข้อมูลไปยังเอาต์พุตสตรีมและปิดสตรีม

โปรแกรมนี้เขียนไปยังเอาต์พุตสตรีมโดยใช้เมธอด println() ข้อมูลที่ถูกเขียนไปยังเอาต์พุตสตรีมของฝั่งไคลเอ็นต์ คือ อินพุตของซีจีไอสคริปต์ที่ฝั่งเซิร์ฟเวอร์

```
BufferedReader in = new BufferedReader(
    new InputStreamReader(c.getInputStream()));
String inputLine;

while ((inputLine = in.readLine()) != null)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
System.out.println(inputLine);
in.close();
```

โดยทั่วไปแล้วเมื่อมีการส่งข้อมูลไปยังซีซีไอสคริปต์ ดังเช่นตัวอย่างนี้ สคริปต์จะอ่านข้อมูลแล้วรับเข้ามาประมวลผลแล้วส่งผลลัพธ์กลับ ดังนั้น โปรแกรมที่ติดต่อกับจึงต้องมีการอ่านข้อมูลหลังจากที่เขียนข้อมูลให้ไปแล้ว

เมื่อสั่งให้โปรแกรมนี้ทำงาน โดยส่งอาร์กิวเมนต์ ดังนี้ "Reverse Me" จะได้ผลลัพธ์ดังนี้

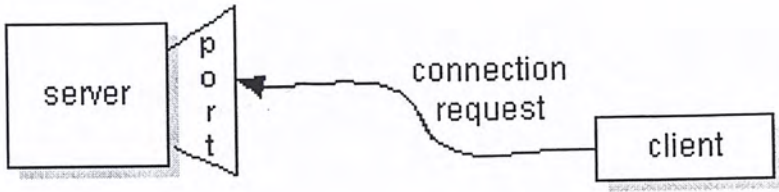
```
Reverse Me
reversed is:
eM esreveR
```

### 2.3.2 ซ็อกเก็ต

นิยาม : ซ็อกเก็ต คือ ปลายทางของการเชื่อมต่อแบบสองทางระหว่างโปรแกรมที่กำลังทำงานอยู่บนเครือข่าย ซ็อกเก็ตจะถูกผูกเข้ากับหมายเลขพอร์ตเพื่อ ที่ซีซีไอสคริปต์ สามารถนำส่งข้อมูลไปยังแอปพลิเคชันปลายทางได้อย่างถูกต้อง

คลาสซูอาร์แอล และ คลาส URLConnections ให้กลไกการทำงานระดับสูงสำหรับการเข้าถึงรีซอร์สบนอินเทอร์เน็ต บางครั้งโปรแกรมอาจต้องการการติดต่อสื่อสารของเครือข่ายระดับล่าง ตัวอย่างเช่น เมื่อต้องการเขียน ไคลเอ็นต์-เซิร์ฟเวอร์แอปพลิเคชัน

ในไคลเอ็นต์-เซิร์ฟเวอร์แอปพลิเคชัน เซิร์ฟเวอร์จะทำหน้าที่ให้บริการต่าง ๆ ไคลเอ็นต์จะเรียกใช้บริการที่เซิร์ฟเวอร์ให้บริการ การติดต่อสื่อสารระหว่างไคลเอ็นต์-เซิร์ฟเวอร์ต้องเชื่อถือได้ นั่นคือ ข้อมูลจะตกหล่นหรือสูญหายระหว่างทางไม่ได้ และ ข้อมูลต้องมาถึงฝ่ายรับตามลำดับที่ถูกส่งออกมาจากฝ่ายส่ง โพรโทคอลที่ซีซีไอให้การเชื่อมต่อที่เชื่อถือได้ ซึ่งโพรโทคอลนี้เป็นโพรโทคอลที่ไคลเอ็นต์-เซิร์ฟเวอร์ใช้เพื่อติดต่อสื่อสารผ่านเครือข่ายอินเทอร์เน็ต ในการติดต่อสื่อสารโดยใช้โพรโทคอลที่ซีซีไอ โปรแกรมไคลเอ็นต์และโปรแกรมเซิร์ฟเวอร์จะสร้างการเชื่อมต่อขึ้นมา แต่ละฝ่ายจะเชื่อมต่อซ็อกเก็ตเข้ากับแต่ละปลายทางของการเชื่อมต่อที่สร้างขึ้นมา ในการติดต่อสื่อสาร ไคลเอ็นต์และเซิร์ฟเวอร์จะเขียนไปยังซ็อกเก็ตและอ่านจากซ็อกเก็ตที่ถูกเชื่อมต่อเข้ากับการเชื่อมต่อ



รูปที่ 2-7 แสดงการร้องขอของไคลเอ็นต์ไปยังเซิร์ฟเวอร์

ที่ฝั่งไคลเอ็นต์ : ไคลเอ็นต์ต้องรู้ชื่อโฮสต์ของเครื่องที่เป็นเซิร์ฟเวอร์และหมายเลขพอร์ตที่ต่ออยู่กับการเชื่อมต่อ ในการสร้างการร้องขอการเชื่อมต่อ ไคลเอ็นต์จะติดต่อกับเซิร์ฟเวอร์ได้โดยระบุชื่อโฮสต์และหมายเลขพอร์ต

ถ้าทุกชั้นตอนดำเนินไปด้วยความเรียบร้อย เซิร์ฟเวอร์จะได้รับการร้องขอการเชื่อมต่อ เมื่อเซิร์ฟเวอร์รับการเชื่อมต่อ เซิร์ฟเวอร์จะสร้างซ็อกเก็ตขึ้นมาใหม่แล้วผูกเข้ากับพอร์ตใหม่เพื่อให้บริการกับไคลเอ็นต์ที่ร้องขอเข้ามา โดยจะเก็บพอร์ตเดิมเพื่อรอคอยการร้องขอจากไคลเอ็นต์ตัวอื่น ๆ



รูปที่ 2-8 แสดงการเชื่อมต่อระหว่างไคลเอ็นต์และเซิร์ฟเวอร์

บนฝั่งไคลเอ็นต์ การเชื่อมต่อจะสำเร็จเมื่อซ็อกเก็ตถูกสร้างขึ้นใหม่แล้ว และ ไคลเอ็นต์สามารถใช้ซ็อกเก็ตเพื่อทำการติดต่อกับเซิร์ฟเวอร์ได้ สังเกตว่าซ็อกเก็ตที่ฝั่งไคลเอ็นต์จะไม่ถูกผูกเข้ากับหมายเลขพอร์ตที่ใช้เพื่อการติดต่อกับเซิร์ฟเวอร์

ตอนนี้ไคลเอ็นต์และเซิร์ฟเวอร์สามารถทำการอ่านจากซ็อกเก็ตและเขียนไปยังซ็อกเก็ตได้แล้ว

แพ็คเกจ java.net ในจาวาแพลตฟอร์มมีคลาสหนึ่ง คือ คลาส Socket ที่อิมพลิเมนต์ฝั่งหนึ่งของการเชื่อมต่อแบบสองฝั่งระหว่างโปรแกรมที่ผู้เขียน เขียนขึ้นกับโปรแกรมอื่น ๆ บนเครือข่าย คลาส Socket จะซ่อนรายละเอียดของระบบทำให้ผู้เขียนไม่ต้องไปยุ่งเกี่ยวกับระบบมากนัก

นอกจากนี้ในแพ็คเกจ java.net ยังมีคลาส ServerSocket ซึ่งอิมพลิเมนต์ซ็อกเก็ตที่เซิร์ฟเวอร์สามารถใช้รอคอยการร้องขอจากไคลเอ็นต์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอ่านจากซ็อกเก็ตและการเขียนไปยังซ็อกเก็ต

ต่อไปนี้เป็นตัวอย่างการเขียนโปรแกรมที่สร้างการเชื่อมต่อไปยังโปรแกรมเซิร์ฟเวอร์โดยใช้คลาสซ็อกเก็ต และอธิบายวิธีการส่งข้อมูลไปยังเซิร์ฟเวอร์และรับข้อมูลจากเซิร์ฟเวอร์ผ่านทางซ็อกเก็ต

ในตัวอย่างนี้ โปรแกรมไคลเอ็นต์ ชื่อ EchoClient เชื่อมต่อไปยังโปรแกรมเซิร์ฟเวอร์ ชื่อ EchoServer โปรแกรมเซิร์ฟเวอร์นี้คอยรับข้อมูลจากไคลเอ็นต์แล้วส่งข้อมูลกลับไปยังไคลเอ็นต์โดยใช้พอร์ตหมายเลข 7

โปรแกรม EchoClient จะสร้างซ็อกเก็ตขึ้นมาเพื่อเชื่อมต่อไปยังโปรแกรม EchoServer โดยโปรแกรม EchoClient จะรับอินพุตจากผู้ใช้ผ่านทางอินพุทสตรีมมาตรฐาน แล้วจะเขียนออกไปยังเซิร์ฟเวอร์ผ่านทางซ็อกเก็ต โปรแกรมเซิร์ฟเวอร์จะส่งข้อมูลกลับมายังไคลเอ็นต์ผ่านทางซ็อกเก็ต โปรแกรมไคลเอ็นต์จะอ่านข้อมูลและแสดงผลข้อมูลที่รับกลับมาจากเซิร์ฟเวอร์

สังเกตว่า EchoClient จะทำหน้าที่ทั้งอ่านข้อมูลจากซ็อกเก็ตและเขียนข้อมูลไปยังซ็อกเก็ต

```
echoSocket = new Socket("taranis", 7);
out = new PrintWriter(echoSocket.getOutputStream(), true);
in = new BufferedReader(new InputStreamReader(echoSocket.getInputStream()));
```

ส่วนของโปรแกรมนี้จะสร้างวัตถุซ็อกเก็ตขึ้นมาและตั้งชื่อว่า *echoSocket* เชื่อมต่อระหว่างไคลเอ็นต์กับเซิร์ฟเวอร์ และเปิดอินพุทสตรีมและเอาต์พุทสตรีมไปของซ็อกเก็ตที่สร้างขึ้นมา ใส่ชื่อโฮสต์และหมายเลขพอร์ตที่ต้องการติดต่อให้กับคอนสตรัคเตอร์ของคลาส *Socket* จากตัวอย่างนี้โปรแกรมใช้โฮสต์ชื่อว่า "taranis" ซึ่งเป็นชื่อที่สมมติขึ้นมา เมื่อจะนำโปรแกรมนี้ไปใช้งานต้องเปลี่ยนชื่อโฮสต์เป็นชื่อที่มีอยู่จริงบนเครือข่าย

ในการส่งข้อมูลผ่านซ็อกเก็ตไปยังเซิร์ฟเวอร์ โปรแกรม EchoClient ต้องเขียนออกไปทาง *PrintWriter* ในการรับการตอบกลับจากเซิร์ฟเวอร์ EchoClient ต้องอ่านจาก *BufferedReader*

```
String userInput;

while ((userInput = stdIn.readLine()) != null) {
    out.println(userInput);
    System.out.println("echo: " + in.readLine());
}
```

ส่วนของโปรแกรมส่วนนี้ จะทำการรับอินพุทของยูสเซอร์ผ่านทางอินพุทมาตรฐาน แล้วเขียนออกไปยังเซิร์ฟเวอร์ผ่านซ็อกเก็ต และอ่านข้อมูลที่ส่งกลับมานำมาแสดงผลออกทางเอาต์พุทมาตรฐาน รูป while จะทำงานวนรอบไปเรื่อย ๆ จนกว่ายูสเซอร์จะพิมพ์ end-of-point เพื่อออกจากลูปและสิ้นสุดการทำงาน

```
out.close();
```

```
in.close();
```

```
stdIn.close();
```

```
echoSocket.close();
```

โปรแกรมที่ดีควรมีการจัดการกับซ็อกเก็ตหลังจากใช้งานแล้ว โปรแกรมส่วนนี้จะทำการปิดสตรีมต่าง ๆ และ ซ็อกเก็ตต่าง ๆ ที่ได้ทำการเปิดเอาไว้ ลำดับในการปิดมีความสำคัญ ควรจะปิดสตรีมที่ต่อเข้ากับซ็อกเก็ตก่อนที่จะปิดซ็อกเก็ต

ตัวอย่างนี้เป็นตัวอย่างโปรแกรมไคลเอ็นต์-เซิร์ฟเวอร์อย่างง่าย ๆ เพราะว่าโพรโตคอลที่ใช้ในการติดต่อง่าย แต่ถ้าผู้เขียนโปรแกรมต้องการเขียนโปรแกรมโดยใช้โพรโตคอลที่มีคสามซับซ้อนขึ้น โปรแกรมไคลเอ็นต์ก็จะต้องมีความซับซ้อนขึ้นตามไปด้วย แต่โดยทั่ว ๆ ไปแล้วหลักการพื้นฐานก็จะเหมือน ๆ กัน ดังนี้

1. เปิดซ็อกเก็ต
2. เปิดอินพุทสตรีมและเอาต์พุทสตรีมไปยังซ็อกเก็ตนั้น
3. อ่านจากสตรีม และ เขียนไปยังสตรีมตามโพรโตคอลที่กำหนดไว้
4. ปิดสตรีม
5. ปิดซ็อกเก็ต

## 2.4 เชนทีทีพี (HTTP)

ข้อมาจาก ไฮเปอร์เท็กซ์ทรานส์เฟอร์โพรโตคอล (HyperText Transfer Protocol) คือโพรโตคอล (Protocol) สำหรับเครือข่ายที่ใช้เพื่อส่งไฟล์และข้อมูลต่าง ๆ บน เวิลด์ไวด์เว็บ (World Wide Web) ไม่ว่ามันจะเป็นไฟล์เอชทีเอ็มแอล (HTML) ไฟล์รูปภาพ หรือไฟล์อะไรก็ตาม โพรแกรมบราวเซอร์ คือ เชนทีทีพี-ไคลเอ็นต์ (HTTP-Client) เพราะว่ามันส่งคำร้องขอต่าง ๆ ไปยัง เชนทีทีพี-เซิร์ฟเวอร์ (HTTP-Server) หรือ ก็คือ โพรแกรมเว็บเซิร์ฟเวอร์ (Web Server) ที่ทำหน้าที่ให้บริการไคลเอ็นต์ โดยคอยส่งการตอบสนองตามที่ไคลเอ็นต์ต้องการ การติดต่อระหว่าง เชนทีทีพี-ไคลเอ็นต์ กับ เชนทีทีพี-เซิร์ฟเวอร์ จะทำผ่านพอร์ต (Port) มาตรฐานของ เชนทีทีพี-เซิร์ฟเวอร์ คือ พอร์ต 80

เชนทีทีพี ใช้โอนย้าย ทรัพยากรต่าง ๆ (resources) ไม่เฉพาะไฟล์เท่านั้น ซึ่งทรัพยากร คือ กลุ่มของข้อมูลทีระบุตำแหน่งด้วย ยูอาร์แอล (URL) รูปแบบที่ธรรมดาที่สุดของทรัพยากร คือ ไฟล์ แต่ทรัพยากรก็ไม่ใช่มีแค่ไฟล์อย่างเดียวอาจจะเป็นผลลัพธ์จาก ซีจีไอ สคริปต์ (CGI Script) ก็ได้ ขณะที่ทำการศึกษารื่องเกี่ยวกับ เชนทีทีพี ถ้าเรามอง ทรัพยากร เป็น ไฟล์จะทำให้เข้าใจได้ง่ายขึ้น

### โครงสร้างของเชนทีทีพี

เชนทีทีพี ก็เหมือนกับโพรโตคอลสำหรับเครือข่ายอื่น ๆ ที่ใช้ รูปแบบไคลเอ็นต์-เซิร์ฟเวอร์ : เชนทีทีพี-ไคลเอ็นต์ สร้างการเชื่อมต่อไปยัง เชนทีทีพี-เซิร์ฟเวอร์ แล้วส่ง การร้องขอบริการ ไปยังเซิร์ฟเวอร์ เมื่อเซิร์ฟเวอร์ได้รับการร้องขอก็จะส่ง การตอบรับกลับไปยัง ไคลเอ็นต์ ซึ่งโดยปกติแล้วก็จะส่งกลับไปพร้อมกับสิ่งที่ไคลเอ็นต์ร้องขอมา เช่น ถ้าไคลเอ็นต์ต้องการไฟล์ index.html ก็จะส่งการร้องขอไปยังเซิร์ฟเวอร์ แล้วเซิร์ฟเวอร์ก็จะส่งการตอบสนองกลับมาพร้อมกับไฟล์ index.html ที่ไคลเอ็นต์ต้องการ หลังจากนำส่งสิ่งที่ไคลเอ็นต์ต้องการแล้ว เซิร์ฟเวอร์ก็จะปิดการเชื่อมต่อ ( ซึ่งการทำงานในลักษณะนี้ทำให้เชนทีทีพีกลายเป็นโพรโตคอลที่ไม่มีสถานะ เพราะ เมื่อไคลเอ็นต์ขอมมา เซิร์ฟเวอร์ก็ให้ ไปแล้วก็เลิกการติดต่อ ทำให้ไม่มีการเก็บสถานะไว้ว่าทำอะไรไปบ้างแล้ว ) รูปแบบของ การร้องขอ และ การตอบรับ อยู่ในรูปแบบที่คล้ายกันและเป็นภาษาอังกฤษทั้งคู่ ซึ่งทั้งคู่ประกอบด้วย :-

- บรรทัดเริ่มต้น
- บรรทัดเฮดเดอร์จะมีหลายบรรทัดหรือไม่เลยก็ได้
- บรรทัดว่าง ( ทั้ง LF และ CR)
- บรรทัดข้อมูลจะมีหรือไม่ก็ได้

รูปแบบของแมสเสจ (message) ของเชนทีทีพีอยู่ในรูปแบบ ดังนี้

< บรรทัดเริ่มต้น > จะแตกต่างกันขึ้นอยู่กับว่าเป็น การร้องขอ หรือ การตอบสนอง

< บรรทัดเซกเตอร์ > มีได้หลายบรรทัด หรือไม่มีเลยก็ได้

Header1: value1

Header2: value2

Header3: value3

← บรรทัดว่าง

< บรรทัดข้อมูล >

บรรทัดเริ่มต้น และ บรรทัดเซกเตอร์ควรจะจบด้วย CRLF ซึ่งค่า แอสกี ( ASCII ) ของ CR และ LF คือ 13 และ 10 ตามลำดับ

บรรทัดเริ่มต้น ของการร้องขอ

บรรทัดเริ่มต้นของการร้องขอจะแตกต่างกันขึ้นอยู่กับว่าใช้วิธีการร้องขอแบบใด บรรทัดเริ่มต้นของการร้องขอแบ่งออกได้เป็น 3 ส่วน ดังนี้ วิธีการร้องขอ พாதของทรัพยากรที่ร้องขอ และ เวอร์ชันของเอชทีทีพี ที่ใช้ ซึ่งแต่ละส่วนแยกออกจากกันด้วยช่องว่าง ตัวอย่างเช่น

GET /path/to/file/index.html HTTP/1.0

ซึ่งเป็นการร้องขอด้วยวิธี GET ( วิธีการร้องขอต้องเป็นตัวพิมพ์ใหญ่เสมอ ) ร้องขอไฟล์ index.html ซึ่งส่วนนี้ “/path/to/file/index.html” จะเรียกว่า URI และ ใช้เอชทีทีพีเวอร์ชัน 1.0 เอชทีทีพีต้องเป็นตัวพิมพ์ใหญ่

บรรทัดเริ่มต้นของการตอบสนอง

บรรทัดเริ่มต้นของการตอบสนอง มักนิยมเรียกว่า บรรทัดแสดงสถานะ ( status line ) แบ่งออกได้เป็น 3 ส่วน ดังนี้ เวอร์ชันของเอชทีทีพี รหัสสถานะการตอบสนอง และ คำอธิบายรหัสสถานะ ตัวอย่างเช่น

HTTP/1.0 200 OK

หรือ

HTTP/1.0 404 Not Found

เวอร์ชันของ เอชทีทีพีต้องเป็นตัวพิมพ์ใหญ่ รหัสสถานะเป็นเลข จำนวนเต็ม 3 ตัว เราสามารถแยกประเภทการตอบสนองได้ตามตัวเลขตัวหน้า ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1XX indicates an informational message only
- 2XX แสดงความสำเร็จชนิดต่าง ๆ
- 3XX เชื่อมต่อไคลเอนต์ไปยังอีก ยูอาร์แอล
- 4XX แสดงความผิดพลาดต่าง ๆ ที่เกิดขึ้นที่ฝั่งไคลเอนต์
- 5XX แสดงความผิดพลาดต่าง ๆ ที่เกิดขึ้นที่ฝั่งเซิร์ฟเวอร์

ตัวอย่างเช่น

#### 200 OK

การร้องขอเป็นผลสำเร็จ และ ข้อมูลที่ต้องการอยู่ในส่วนบรรทัดข้อมูล

#### 404 NOT FOUND

ไม่พบยูอาร์แอลที่ร้องขอ

#### 301 Moved Permanently

#### 302 Moved Temporarily

#### 303 See Other (HTTP 1.1 only)

ทรัพยากรที่ร้องขอถูกย้ายไปอีกยูอาร์แอล แสดงไว้ในบรรทัดเฮดเดอร์

#### 500 server error

เซิร์ฟเวอร์ทำงานผิดพลาด

#### บรรทัดเฮดเดอร์

บรรทัดเฮดเดอร์ให้ข้อมูลและรายละเอียดต่าง ๆ เกี่ยวกับการร้องขอ และการตอบสนอง หรือเกี่ยวกับสิ่งที่อยู่ใน บรรทัดข้อมูล บรรทัดเฮดเดอร์จะมีได้หลายบรรทัด แต่ละบรรทัดของเฮดเดอร์จะอยู่ในรูป

"Header-Name: value" และจบด้วย CRLF ชื่อของเฮดเดอร์ จะเป็นตัวเล็กหรือตัวใหญ่ก็ได้ บรรทัดเฮดเดอร์ที่เริ่มด้วยช่องว่าง หรือ แท็บ จะถือว่าเป็นของเฮดเดอร์บรรทัดก่อนหน้านี้ เช่น

**Header1: some-long-value-1a, some-long-value-1b**

มีค่าเท่ากับ

**Header1:   some-long-value-1a,  
          some-long-value-1b**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ตัวอย่าง เฮดเดอร์

**From :** แสดง E-mail Address ของใครก็ตามที่ทำการร้องขอ

**User-Agent :** แสดงโปรแกรมที่ทำการร้องขอ อยู่ในรูปแบบดังนี้ **Program-name/x.xx** ซึ่ง X.XX คือเวอร์ชันของโปรแกรม ตัวอย่างเช่น Netscape 3.0 จะส่งเฮดเดอร์ดังนี้ **"User-agent: Mozilla/3.0Gold"**.

เฮดเดอร์ ช่วยให้ผู้ดูแลเว็บ ( webmaster ) แก้ปัญหาได้

### บรรทัดข้อมูล

เอชทีทีพีแมสเชจ อาจจะมีหรือไม่มีส่วนนี้ก็ได้อ สำหรับการตอบสนอง ส่วนนี้เป็นส่วนที่ทรัพยากรที่ร้องขอถูกส่งกลับมายังไคลเอ็นต์ สำหรับการร้องขอ ส่วนนี้เป็นส่วนที่ใส่ไฟล์ที่ต้องการจะอัปโหลด ( upload ) ไปยังเซิร์ฟเวอร์

ถ้า เอชทีทีพีแมสเชจ มีส่วน บรรทัดข้อมูล ก็จะมีส่วนเฮดเดอร์ที่อธิบายส่วนข้อมูลด้วย เช่น

- **Content-Type:** เป็นเฮดเดอร์ที่บอก MIME-type ของข้อมูลที่อยู่ในบรรทัดข้อมูล ว่าเป็นไฟล์ชนิดอะไร เช่น text/html หรือ image/gif.
- **Content-Length:** เป็นเฮดเดอร์ที่บอกจำนวนไบต์ของข้อมูลที่อยู่ในบรรทัดข้อมูล

### ตัวอย่างการติดต่อสื่อสารกันด้วยโพรโตคอลเอชทีทีพี

ถ้าเราต้องการไฟล์ที่ระบุด้วย <http://www.somehost.com/path/file.html>

ขั้นแรกเราต้องเปิดซ็อกเก็ต ( socket ) ไปยังโฮสต์ [www.somhost.com](http://www.somhost.com) ที่พอร์ต 80 แล้วส่งข้อความต่อไปนี้ไปตามซ็อกเก็ต

```
GET /path/file.html HTTP/1.0
From: someuser@jmarshall.com
User-Agent: HTTPTool/1.0
[blank line here]
```

### เซิร์ฟเวอร์จะตอบสนองมาดังนี้

```
HTTP/1.0 200 OK
Date: Fri, 31 Dec 1999 23:59:59 GMT
Content-Type: text/html
Content-Length: 1354
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<html>
<body>
<h1>Happy New Millennium!</h1>
(more file contents)
.
.
.
</body>
</html>

```

หลังจากไคลเอ็นต์ได้รับการตอบสนองมาเรียบร้อยแล้วเซิร์ฟเวอร์จะปิดชอกร์กีด นอกจากการร้องขอด้วยวิธี GET แล้ว โพรโตคอลเอชทีทีพียังมีวิธีการร้องขอแบบอื่น ๆ อีกเช่น วิธี POST และ HEAD

### HEAD

การร้องขอด้วยวิธีนี้ จะเหมือนกับการร้องขอด้วยวิธี GET แต่ต่างกันตรงที่วิธีนี้จะขอให้เซิร์ฟเวอร์ส่งมาเฉพาะส่วนเฮดเดอร์เท่านั้น วิธีนี้มีประโยชน์มากในการตรวจสอบทรัพยากรที่เราต้องการ โดยไม่ต้องดาวน์โหลดไฟล์นั้นมา ซึ่งจะช่วยประหยัดแบนด์วิดธ์ วิธีนี้จะใช้เมื่อไม่ต้องการเนื้อหาของไฟล์ การตอบสนองของการร้องขอด้วยวิธีนี้จะไม่มีส่วนของบรรทัดข้อมูล จะมีก็เพียงบรรทัดเริ่มต้น และ บรรทัดเฮดเดอร์

### POST

การร้องขอด้วยวิธีนี้ใช้ส่งข้อมูลไปยังเซิร์ฟเวอร์เพื่อเอาไปประมวลผลบางอย่าง การร้องขอด้วยวิธี POST แตกต่างจากวิธี GET บางประการดังนี้

- วิธี POST จะมีข้อมูลถูกส่งไปกับการร้องขอซึ่งข้อมูลจะอยู่ในส่วนบรรทัดข้อมูล และจะมีเฮดเดอร์พิเศษอธิบายสิ่งที่อยู่ในบรรทัดข้อมูล เช่น Content-Type: และ Content-Length:.
- *URI* ที่ระบุอยู่ในการร้องขอไม่ใช่ทรัพยากรที่จะนำเอามาได้ โดยทั่วไปแล้วจะระบุตำแหน่งของโปรแกรมที่ทำหน้าที่คอยจัดการกับข้อมูลที่ถูกส่งไป
- ส่วนใหญ่แล้วการตอบสนองของการร้องขอวิธีนี้จะเป็นผลลัพธ์ที่ได้จากการประมวลผลข้อมูลที่ส่งเข้ามามากกว่าที่จะเป็นไฟล์ที่มีอยู่แล้ว

การใช้งานทั่วไปของการร้องขอด้วยวิธีนี้คือ การส่งข้อมูลที่กรอกผ่านฟอร์มของไฟล์เอชทีเอ็มแอลไปยัง ซีจีไอ สคริปต์ ในกรณีนี้ ส่วน Content-Type:จะเป็น application/x-www-form-urlencoded และ Content-Length: จะเป็นความยาวของข้อมูล ซีจีไอสคริปต์จะรับข้อมูลเข้าผ่านทางอินพุตมาตรฐาน ตัวอย่างเช่น

```
POST /path/script.cgi HTTP/1.0
From: frog@jmarshall.com
User-Agent: HTTPTool/1.0
Content-Type: application/x-www-form-urlencoded
Content-Length: 32
```

เราสามารถใช้อ้างอิงด้วยวิธี POST เพื่อส่งข้อมูลที่ต้องการจะส่งไม่จำกัดแต่เพียงการกรอกข้อมูลผ่านฟอร์มเท่านั้น แต่ยังสามารถส่งไฟล์ต่าง ๆ ได้อีกด้วย แต่เราต้องแน่ใจว่าฝ่ายรับรู้จักรูปแบบของไฟล์ที่ส่งไป

การร้องขอด้วยวิธี GET ก็สามารถส่งข้อมูลผ่านฟอร์มได้ด้วยเช่นกัน ข้อมูลจะผ่านการเข้ารหัสแบบ URL-encoded แล้วต่อท้าย URI ไป

## 2.5 หลักการพื้นฐานของอาร์เอ็มไอ (RMI)

### 2.5.1 รีโมตโพรซีเจอร์คอลล์ (Remote Procedure Call - RPC)

รีโมตโพรซีเจอร์คอลล์ เป็นวิธีการในการส่งข้อมูลหรือเรียกบริการจากเครื่องที่อยู่ในระยะไกลด้วยเทคโนโลยีที่ไม่เป็นวิธีเชิงวัตถุ

จุดประสงค์ของการทำรีโมตโพรซีเจอร์คอลล์เหมือนกับการทำรีโมตเมธอดอินโวนเคชัน คือเพื่อสร้างการส่งข้อมูลไปยังโฮสต์ที่อยู่ในระยะไกล โดยการทำรีโมตโพรซีเจอร์คอลล์นั้นแสดงการส่งข้อมูลระยะไกลในรูปแบบของการเรียกใช้โพรซีเจอร์ ซึ่งเป็นวิธีพื้นฐานของการส่งข้อมูลแบบที่ไม่ใช่วิธีเชิงวัตถุ ซึ่งแตกต่างจากวิธีการเรียกใช้เมธอด คือ อาร์กิวเมนต์ (argument) และค่าที่จะถูกส่งกลับของโพรซีเจurnั้นเป็นประเภทพื้นฐานทั่วไป ไม่ได้เป็นประเภทวัตถุ ส่วนการทำรีโมตเมธอดอินโวนเคชันนั้น มีเงื่อนไขพื้นฐานเหมือนกับการทำรีโมตโพรซีเจอร์คอลล์ แต่ได้มีการเพิ่มรายละเอียดและความสามารถ โดยการอนุญาตให้อาร์กิวเมนต์และค่าที่ถูกส่งกลับเป็นประเภทของวัตถุได้

ตัวอย่างของการทำรีโมตโพรซีเจอร์คอลล์ดังนี้  $\text{float side} = \text{sqrt}(\text{area});$

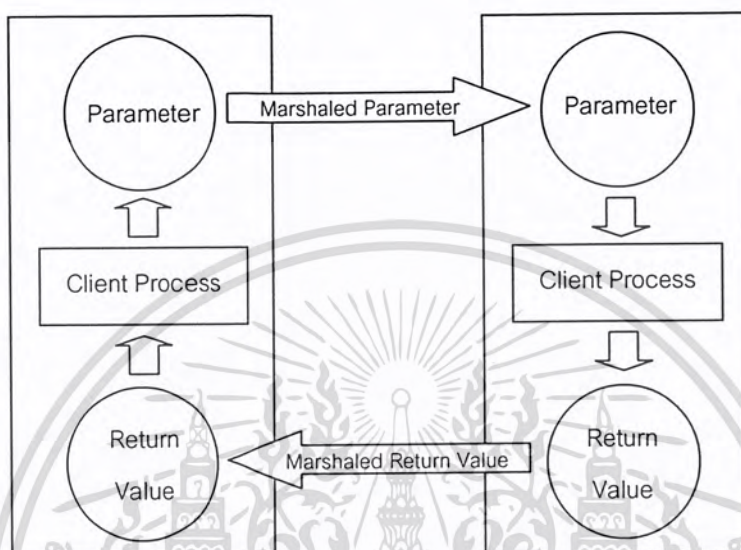
การเรียกใช้โพรซีเจอร์ sqrt โดยส่งค่า area เป็นอาร์กิวเมนต์ และเก็บค่าที่ส่งกลับมาจากโพรซีเจอร์ sqrt เก็บไว้ในตัวแปร side ซึ่งทำตามขั้นตอน ดังต่อไปนี้

1. นำแอดเดรสของโฮสต์ระยะไกลออกมา
2. เปิดการเชื่อมต่อซ็อกเก็ตไปยังฝั่งเซิร์ฟเวอร์
3. แปลงค่าที่ต้องการนำไปคำนวณตามการทำงานของโพรซีเจอร์ ให้เป็นแอสกี (ส่งเป็นไบนารี)
4. เมื่อฝ่ายร้องขอใช้โพรซีเจอร์ได้รับการตอบสนองจากเซิร์ฟเวอร์ จะทำการอ่าน และแปลงค่าจากแอสกี ให้เป็นเลขทศนิยม

สิ่งที่ยุ่งยากที่เกิดจากการเรียกใช้โพรซีเจอร์ระยะไกลคือ การแปลงค่าของอาร์กิวเมนต์จากค่าทศนิยมเป็นแอสกี เรียกกระบวนการนี้ว่าการมาร์แชลลิง(marshalling)อาร์กิวเมนต์ การแปลงค่าที่ส่งกลับจากแอสกี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นค่าทศนิยมนั้น เรียกว่า กระบวนการอันมาร์แชลลิง(unmarshalling)ค่าที่ส่งกลับ เซิร์ฟเวอร์ต้องทำกระบวนการเหล่านี้ได้อย่างสมบูรณ์ นั่นคือ ต้องมีการแปลงค่าอาร์กิวเมนต์(กระบวนการอันมาร์แชลลิง) เพื่อทำการคำนวณตามการทำงานของโปรแกรมเมอร์ และทำตามกระบวนการมาร์แชลลิง เพื่อส่งค่ากลับไปให้ผู้เรียกใช้ ดังรูปที่ 2-9



รูปที่ 2-9 แสดงการมาร์แชลลิงและอันมาร์แชลลิง

## 2.5.2 รีโมตเมธอดอินโวกชัน(Remote Method Invocation - RMI)

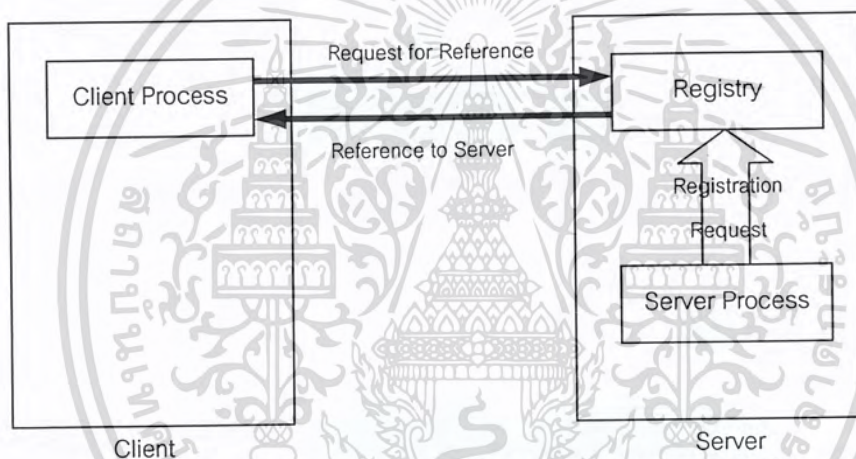
รีโมตเมธอดอินโวกชัน เป็นวิธีที่โปรแกรมเมอร์ (ที่ใช้ภาษาจาวาในการเขียน) สามารถส่งข้อมูลหรือเรียกบริการจากเครื่องที่อยู่ในระยะไกลโดยวิธีเชิงวัตถุได้ โดยวัตถุบนเครื่องคอมพิวเตอร์ต่าง ๆ สามารถติดต่อผ่านเครือข่ายโดยวิธีกระจายได้ รีโมตเมธอดอินโวกชันเป็นรีโมตโปรแกรมเมอร์คอลล์ในรูปแบบของภาษาจาวา แต่ด้วยความสามารถที่ส่งวัตถุจำนวนหนึ่งหรือมากกว่านั้นได้ เมื่อมีการร้องขอการบริการจากเครื่องระยะไกล ซึ่งวัตถุนั้น ๆ สามารถรวบรวมข่าวสารที่จะใช้ในการให้บริการซึ่งเกิดขึ้นบนเครื่องคอมพิวเตอร์ระยะไกล บริษัทซันไมโครซิสเต็มส์ ผู้ให้กำเนิดภาษาจาวา ได้เรียกสิ่งนี้ว่า มูฟวิงบีแฮฟวิเออร์ (moving behavior) จุดประสงค์ของจาวา รีโมตเมธอดอินโวกชัน คือการจัดเตรียมเฟรมเวิร์กสำหรับวัตถุ ให้สามารถสื่อสารถึงกัน โดยผ่านเมธอดของมันเอง ซึ่งหมายความว่า ไคลเอ็นต์สามารถเข้าถึงเซิร์ฟเวอร์บนฝั่งไคลเอ็นต์หรือบนเครือข่ายได้ ถ้าหากกำลังทำงานอยู่ในระบบรันไทม์เดียวกัน

การทำรีโมตเมธอดอินโวกชันดีกว่าการทำรีโมตโปรแกรมเมอร์คอลล์ โดยมีการทำกระบวนการมาร์แชลลิงและกระบวนการอันมาร์แชลลิงวัตถุได้อย่างอัตโนมัติ ในการทำงานลักษณะนี้ ต้องใช้คุณสมบัติซีเรียไลเซชัน (serialization) ของภาษาจาวา คือทำการซีเรียไลซ์อาร์กิวเมนต์และค่าที่ถูกส่งกลับได้ แล้วทำการส่งข้ามเครือข่าย และเมื่อได้รับการตอบสนองจากผู้ให้บริการแล้วมีการทำ ดีซีเรียไลเซชันข้อมูลที่ได้ เพื่อนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานต่อไป ดังนั้นวัตถุทั้งหมดที่ส่งผ่านรีโมตเมธอดอินโวก์ชัน ในรูปแบบของอาร์กิวเมนต์ หรือค่าที่ส่งกลับ จำเป็นต้องอิมพลีเมนต์ (implements) ซีเรียไลเซเบิลอินเทอร์เฟซ (Serializable Interface) ถ้าอาร์กิวเมนต์หรือค่าที่ส่งกลับไม่สามารถถูกซีเรียไลซ์ได้ การเรียกเมธอดระยะไกลจะไม่สามารถเกิดขึ้นได้

การทำรีโมตเมธอดอินโวก์ชันได้จัดให้มีบริการรีจิสทรี(registry) ซึ่งทำงานอยู่ที่ฝั่งเซิร์ฟเวอร์ การบริการรีจิสทรีนั้นจะทำการเก็บรักษาฐานข้อมูลของวัตถุในระยะไกลในเซิร์ฟเวอร์นั้น ๆ ซึ่งได้ทำการลงทะเบียนไว้แล้ว เมื่อไคลเอ็นต์ต้องการใช้บริการวัตถุระยะไกล ฝั่งไคลเอ็นต์จะทำการร้องขอโดยระบุชื่อวัตถุระยะไกลที่ต้องการร้องขอ เพื่อการอ้างอิงถึงวัตถุนั้น ๆ รีจิสทรีจะทำการตรวจสอบฐานข้อมูลของตัวเอง และรายงานตัวอ้างอิง (reference) ถึงวัตถุที่ต้องการเพื่อใช้บริการ ดังรูปแสดงการทำงานของรีจิสทรีตามวิธีการรีโมตเมธอดอินโวก์ชัน



รูปที่ 2-10 แสดงการทำงานของรีจิสทรีตามวิธีการรีโมตเมธอดอินโวก์ชัน

การทำรีโมตเมธอดอินโวก์ชันนั้น วัตถุระยะไกลต้องถูกรีจิสเตอร์ด้วยบริการเนมมิง (naming) บริการนี้อนุญาตให้ฝั่งไคลเอ็นต์ระบุถึงตำแหน่งที่จะทำงาน หรือฝั่งไคลเอ็นต์สามารถดึงเอาตัวอ้างอิงถึงวัตถุระยะไกลได้นั่นเอง โดยไคลเอ็นต์จะติดต่อกับ เนมมิง รีจิสทรี (naming registry) และร้องขอตัวอ้างอิงถึงวัตถุระยะไกลที่ต้องการไปยังผู้ให้บริการด้วยการบอกชื่อของวัตถุนั้น หลังจากนั้นเนมมิงรีจิสทรีจะคืนค่าตัวอ้างอิงวัตถุระยะไกลให้

การสร้างคลาสที่สามารถถูกเข้าถึงได้จากระยะไกลนั้น สิ่งที่ต้องทำคือ อย่างแรกต้องระบุอินเทอร์เฟซที่ใช้ซึ่งประกาศเมธอดที่ต้องการใช้ให้เป็นพับลิก (public) โดยที่พารามิเตอร์ (parameter) หรือค่าที่ส่งกลับนั้นจะเป็นประเภทอะไรก็ได้ คลาสที่สามารถถูกเข้าถึงได้ในระยะไกลนั้นต้องอิมพลีเมนต์อินเทอร์เฟซดังกล่าว รวมทั้งอินเทอร์เฟซและเมธอดอื่น ๆ ที่จำเป็นในการทำงานบนฝั่งที่ร้องขอ และมีการสร้างสแต็บ (stub) และสเกเลตัน (skeleton) โดยคำสั่ง rmic สแต็บเป็นคลาสที่แปลการเรียกเมธอดในระยะไกลต่าง ๆ ให้เป็นการติดตั้งการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สื่อสารของเครือข่ายและการส่งค่าพารามิเตอร์อย่างอัตโนมัติ สเกเลตันเป็นคลาสที่อยู่ฝั่งเซิร์ฟเวอร์ (อยู่ที่ระยะไกล) และรับการเชื่อมต่อของเครือข่ายและแปลให้อยู่ในรูปของการเรียกเมธอดบนวัตถุจริง ๆ

### วัตถุระยะไกล (Remote Object)

วัตถุใด ๆ ที่จัดตั้งขึ้นเพื่อรองรับการเรียกเมธอดจากวัตถุอื่น ๆ ที่ทำงานในระยะไกล ซึ่งประกอบด้วย 2 ส่วน คือ ส่วนอินเทอร์เฟซที่อธิบายเมธอดของวัตถุ และการทำงานของอินเทอร์เฟซนี้

### สตัปและสเกเลตัน (Stub and Skeleton)

รีโมตเมธอดอินโวก์ชันใช้วิธีมาตรฐานสำหรับการติดต่อกับวัตถุในระยะไกล นั่นคือ การใช้สตัปและสเกเลตัน สตัปสำหรับวัตถุระยะไกลนั้น จะทำงานเหมือนเป็นตัวแทนของฝั่งผู้ร้องขอหรือฝั่งไกลเอ็นด์ สตัปจะรับผิดชอบในการทำงานของเมธอดที่ถูกเรียกซึ่งอยู่บนวัตถุระยะไกล สตัปจะอิมพลิเมนต์อินเทอร์เฟซตัวเดียวกับที่วัตถุระยะไกลอิมพลิเมนต์อยู่

เมื่อเมธอดของสตัปถูกเรียก สตัปจะทำงานตามขั้นตอนต่อไปนี้

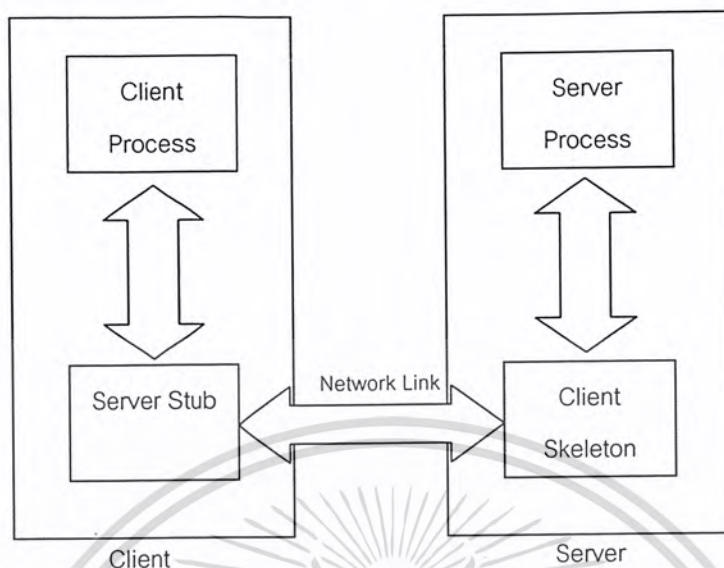
- สร้างการเชื่อมต่อถึงวัตถุระยะไกล
- ทำกระบวนการมาร์แชล (ทำการเขียนและส่ง) พารามิเตอร์ ไปให้วัตถุระยะไกล
- รวบรวมผลลัพธ์ของเมธอดที่ร้องขอ
- ทำการอันมาร์แชล (ทำการอ่าน) ค่าที่ส่งกลับหรือปัญหาที่เกิดขึ้น
- ส่งค่าให้แก่ผู้ร้องขอ

สเกเลตันรับผิดชอบสำหรับการส่งการเรียกไปยังวัตถุในระยะไกลจริง ๆ เมื่อสเกเลตันได้รับการร้องขอใช้เมธอด สเกเลตันต้องทำงานตามขั้นตอนต่อไปนี้

- กระบวนการอันมาร์แชล (ทำการอ่าน) พารามิเตอร์ เพื่อนำไปใช้ในเมธอดของวัตถุระยะไกล
- ร้องขอเมธอดบนการทำงานของวัตถุระยะไกลจริง ๆ
- ทำกระบวนการมาร์แชล (ทำการเขียนและส่ง) ผลลัพธ์ ซึ่งอาจจะเป็นค่าผลลัพธ์หรือปัญหาที่เกิดขึ้นออกไปแก่ผู้ร้องขอ

กล่าวได้ว่า สตัปและวัตถุฝั่งไกลคอล จะอยู่บนฝั่งไกลคอลโฮสต์ และในทำนองเดียวกัน สเกเลตันและวัตถุในระยะไกลจะอยู่ที่โฮสต์ในระยะไกลเช่นกัน โดยวัตถุไกลคอลจะสื่อสารได้โดยอาศัยสตัป และวัตถุในระยะไกลก็สามารถสื่อสารได้โดยอาศัยสเกเลตัน หากไคลเอ็นต์ต้องการเรียกใช้เมธอดบนวัตถุที่อยู่ระยะไกลโดยตรง การเรียกใช้เมธอดนั้นจะถูกส่งผ่านสตัป ซึ่งเก็บรายละเอียดของการติดตั้งการสื่อสาร รวมทั้งส่งการเรียกเมธอดและพารามิเตอร์ไปยังสเกเลตัน ส่วนสเกเลตันนั้นจะสร้างการเรียกเมธอดที่อยู่บนวัตถุที่ต้องการนั้นจริงๆ ค่าที่ส่งกลับเป็นผลลัพธ์ที่ถูกส่งกลับมาจากสเกเลตันไปให้สตัป ซึ่งจะส่งผลลัพธ์ให้กับไคลเอ็นต์ต่อไป ราวกับว่าเป็นการเรียกเมธอดบนโลกคอล(local)เสียเอง ดังแสดงดัง รูปที่ 2-11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-11 แสดงการติดต่อสื่อสารผ่านสแต็บและสเกเลตันของวัตถุโลคอลและวัตถุระยะไกล

### วิธีการเนมมิง (Naming Mechanism)

ไคลเอ็นต์สามารถร้องขอเมธอดบนวัตถุในระยะไกลได้ เพียงแต่มีการอ้างอิงถึงวัตถุระยะไกลนั้น เนมมิงรีจิสทรีถูกจัดเตรียมไว้ในรีโมตเมธอดอินโวก์ชันสำหรับการรองรับการอ้างอิงถึงวัตถุ ซึ่งเนมมิงรีจิสทรีจะเริ่มค้น เมื่อมีการใช้คำสั่ง `rmiregistry` เพื่อลงทะเบียนวัตถุที่ให้บริการจากระยะไกล

ไคลเอ็นต์สามารถดึงเอาตัวอ้างอิงถึงวัตถุระยะไกลผ่านบริการรีจิสทรี โดยใช้เมธอด `lookup` ของคลาสเนมมิง โดยต้องมีการกำหนดยูอาร์แอล (URL) เพื่อระบุถึงชื่อโฮสต์ที่ไคลเอ็นต์จะเข้าไปใช้บริการ จากวัตถุระยะไกลอาจจะกำหนดเป็นชื่อโฮสต์หรือแอดเดรสก็ได้ ซึ่งยูอาร์แอลต้องมีคำว่า `rmi` อยู่ด้วยเป็นโปรโตคอลของมัน เมื่อไคลเอ็นต์ได้ตัวอ้างอิงไปยังวัตถุในระยะไกลแล้ว ก็เสมือนว่าวัตถุนั้นเป็นวัตถุโลคอล

### 2.6 หลักการพื้นฐานของเซรต

ระบบปฏิบัติการแบบมัลติเซรต (multi-threading) จะสามารถแบ่งโปรเซส (process) ออกเป็นหน่วยที่เล็กกว่าซึ่งเรียกว่าเซรต (thread) หรือบางทีเรียกว่า ไลท์เวทโปรเซส (light-weight process) หมายความว่าในหนึ่งโปรเซสอาจมีได้หลายเซรต โดยที่เซรตเหล่านั้นใช้ข้อมูลบางตัวหรือโปรแกรมบางส่วนร่วมกันในโปรเซสนั้น โดยปกติเซรตหนึ่งจะถูกออกแบบให้ทำงานอย่างหนึ่ง และเมื่อนำหลาย ๆ เซรตมารวมกันในโปรเซสหนึ่งจะช่วยทำให้โปรเซสนั้นสามารถทำงานได้มากกว่าหนึ่งอย่างพร้อม ๆ กันโดยไม่ต้องมีการทำคอนเท็กซ์สวิตช์ (context switching) ระหว่างโปรเซส โปรแกรมจึงทำงานได้เร็ว และเป็นการใช้งานหน่วยประมวลผลได้อย่างมีประสิทธิภาพมากขึ้น เช่น หากต้องการโปรแกรมที่ทำการอ่านข้อมูลจากดิสก์ (Disk) ไปพร้อมกับพิมพ์ตัวอักษรบนจอภาพ ก็สร้างสองเซรตโดยให้เซรตแรกทำการอ่านข้อมูลจากดิสก์ และเซรตที่สองทำการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิมพ์ตัวอักษรบนจอภาพ เมื่อให้โปรแกรมทำงานมีการสลับกันทำงานระหว่างเซรตทั้งสองอย่างรวดเร็วจนดูเหมือนกับการทำงานทั้งสองเกิดขึ้นพร้อม ๆ กัน

ในระบบมัลติทาสกิง (multitasking) โดยปกติหนึ่งโพรเซสจะมีหนึ่งพีซีบี (PCB) ส่วนในระบบมัลติเซรต หนึ่งโพรเซสจะมีหนึ่งพีซีบีเช่นกันแต่อาจมีหลายเซรต โดยที่เซรตเหล่านั้นมีส่วนที่เป็นโปรแกรม (code segment) ร่วมกันและใช้หน่วยความจำสำหรับเก็บข้อมูล (data segment) บางส่วนร่วมกัน ดังจะแสดงโครงสร้างหน่วยความจำของโพรเซส

หากเซรตในโพรเซสเดียวกันใช้ตัวแปรบางตัวร่วมกัน เราเรียกตัวแปรเหล่านั้นว่า แชร่เวเรียเบิล (shared variable) และถ้าเซรตหนึ่งมีตัวแปรของมันเองที่ไม่ถูกใช้โดยเซรตอื่น ก็จะเรียกตัวแปรนั้นว่า โลคอลเวเรียเบิล (local variable) โดยปกติเซรตหนึ่งจะทำงานในบางส่วนของโค้ดเซกเมนต์เท่านั้น แต่ละเซรตจะมีโปรแกรมของตัวเองซึ่งมีจุดเริ่มต้นและจุดสิ้นสุดแตกต่างกันไป และเซรตแต่ละตัวมักจะทำงานเป็นอิสระต่อกัน ทำให้เซรตหลาย ๆ ตัวสามารถทำงานพร้อมกันได้ ถ้าเครื่องคอมพิวเตอร์นั้นมีหน่วยประมวลผลหลายตัวก็อาจแบ่งให้หน่วยประมวลผลแต่ละตัวทำงานแต่ละเซรตได้พร้อมกันแบบขนาน (parallel) ซึ่งจะช่วยให้โปรแกรมทำงานเร็วขึ้น

การจัดการให้เซรตใดถูกทำงานเมื่อใด หรือหน่วยประมวลผลใดทำงานเซรตใดนั้นเป็นหน้าที่ของระบบปฏิบัติการ ไม่ได้อยู่ในการควบคุมของผู้เขียนโปรแกรมโดยตรง ที่ผู้เขียนโปรแกรมต้องทำคือการออกแบบและการสร้างเซรตสำหรับหน้าที่นั้น ๆ แล้วสร้างโปรแกรมควบคุมให้เซรตเหล่านั้นเริ่มต้นทำงานและหยุดลงเมื่อสิ้นสุดการทำงาน ในระหว่างที่เซรตทำงานโปรแกรมยังสามารถออกคำสั่งให้เซรตนั้นหยุดและเริ่มต้นทำงานเมื่อถึงเวลาที่กำหนดได้

ข้อแตกต่างระหว่างเซรต กับ โพรเซส พอสรุปได้ดังนี้

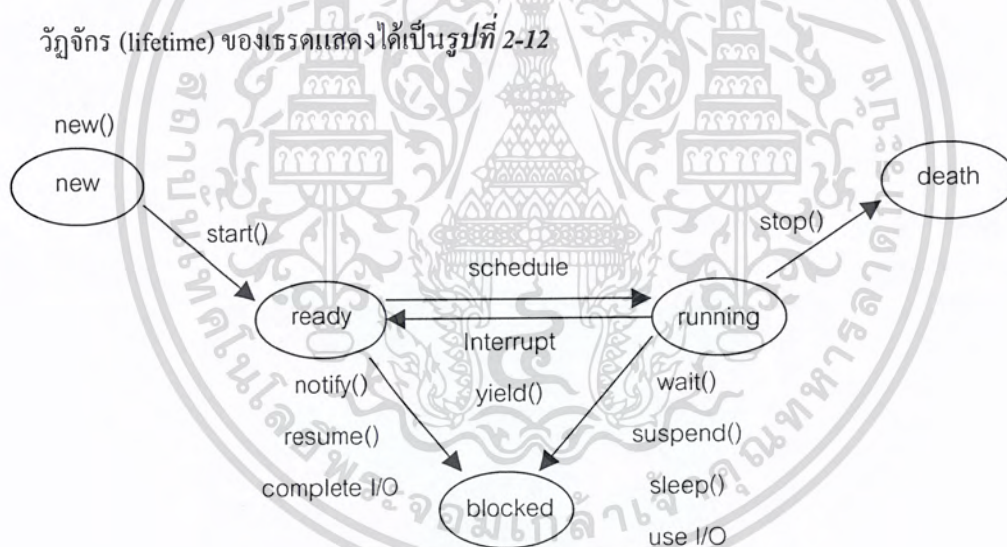
1. Dependency แม้ว่าแต่ละเซรตของโพรเซสเดียวกันจะทำงานที่แตกต่างกัน แต่มีความเกี่ยวข้องกันบางอย่างและต้องทำงานอยู่ในเอ็นไวรอนเมนต์ (environment) เดียวกัน จึงมีการใช้ตัวแปรบางตัวร่วมกัน แต่สำหรับโพรเซสจะไม่มีเกี่ยวข้องกันเลย เพราะแต่ละโพรเซสจะมีเอ็นไวรอนเมนต์ของตนเอง
2. Communication การติดต่อระหว่างเซรตทำได้ง่าย โดยการกำหนดค่าและอ่านค่า แชร่เวเรียเบิล แต่สำหรับโพรเซสจะไม่มีแชร่เวเรียเบิล เนื่องจากเมื่อมีการทำคอนเทกซ์สวิตช์ ก็จะมีการเปลี่ยนเอ็นไวรอนเมนต์ของโพรเซสด้วย วิธีเดียวที่โพรเซสจะส่งค่าให้แก่กันได้คือทำผ่านทางไฟล์ (เรียกว่า ไปได้) ซึ่งเป็นวิธีที่สิ้นเปลืองมากและช้ากว่าการอ่านหรือหรือกำหนดค่าตัวแปรอย่างมาก
3. Context switching เมื่อโพรเซสหนึ่งถูกจัดลำดับเข้าทำงาน หากโพรเซสนั้นมีมากกว่าหนึ่งเซรต ก็จะมีการสลับกันทำงานระหว่างเซรตเหล่านั้น โดยที่ไม่ต้องเปลี่ยนเอ็นไวรอนเมนต์และพีซีบีของโพรเซส แต่จะเปลี่ยนแปลงโลคอลเวเรียเบิลของแต่ละเซรตเท่านั้น ทำให้การสลับการ

ทำงานระหว่างเซรค เป็นภาระต่อเครื่องน้อยกว่าการทำคอนเท็กซ์สวิตช์ของการเปลี่ยนโพรเซส ส่งผลให้เซรคเหล่านั้นมีเวลาทำงานมากขึ้น

### สถานะของเซรคและวัฏจักรของเซรค

สถานะของเซรคแบ่งได้เป็น 5 สถานะ มีรายละเอียดดังนี้

- New คือ ตอนที่เซรคถูกสร้างขึ้นใหม่ ๆ มีการสร้างพื้นที่หน่วยความจำสำหรับตัวแปรในเซรคนั้น (โลคอลเวเรียเบิล)
- Ready คือ ตอนที่เซรคพร้อมที่จะทำงาน และกำลังรอเพื่อให้ถูกเลือกไปทำงาน
- Running คือ ตอนที่เซรคถูกหน่วยประมวลผลทำงาน
- Death คือ ตอนที่เซรคทำงานจบแล้วและรอที่จะถูกทำลาย
- Blocked คือ ตอนที่เซรคกำลังรอเพื่อจะกลับเข้าสู่สถานะ ready ซึ่งเซรคเหล่านี้เคยถูกหน่วยประมวลผลทำงานแล้ว แต่ต้องถูกหยุดลงด้วยเหตุผลบางอย่าง



รูปที่ 2-12 แสดงวัฏจักรของเซรค

การเปลี่ยนแปลงสถานะของเซรค ในภาษาจาวามีกฎเกณฑ์ดังนี้

1. โปรแกรมหนึ่งอาจสร้างเซรคหนึ่งขึ้นมาได้โดยการ new instance ของคลาสเซรค เมื่อเซรคถูกสร้างขึ้นแล้ว มันจะเข้าสู่สถานะ new เพื่อทำการกำหนดค่าเริ่มต้นต่าง ๆ
2. เมื่อเซรคที่อยู่สถานะ new ได้รับคำสั่ง start() จากโปรแกรม มันจะเปลี่ยนไปสู่สถานะ ready และจะรออยู่ในคิวของเซรค จนกว่าจะถูกเลือกออกไปทำงานในหน่วยประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. หากไม่มีเชรคในสถานะ running ระบบปฏิบัติการจะทำการ schedule คือเลือกเชรคหนึ่งจากคิวของเชรคที่รออยู่ในสถานะ ready เพื่อไปทำงาน กฎเกณฑ์ที่ใช้ในการ schedule นี้ขึ้นกับระบบปฏิบัติการที่ใช้งาน (ไม่มีการกำหนดกฎเกณฑ์ในภาษาจาวา) เช่นในระบบ Solaris 1.x จะใช้ไพอรอริตี้ (priority) ของเชรคเป็นตัวเลือก ส่วนในระบบ Win32 จะใช้ทั้งไพอรอริตี้ กับ ไทม์สไลซ์ (timeslice) ของเชรคจัดการเลือก
4. เมื่อเชรคเข้ามาอยู่ในสถานะ running จะเปลี่ยนสถานะต่อไป 3 แบบ คือ
  - 4.1 เป็นสถานะ death เมื่อเชรคนั้นได้รับคำสั่ง stop() จากโปรแกรมเชรค ที่อยู่ในสถานะ death จะถูกทำการเพื่อคืนพื้นที่หน่วยความจำและไม่สามารถกลับเข้าสู่สถานะ ready ได้อีก
  - 4.2 เป็นสถานะ ready เมื่อเชรคนั้นถูกอินเทอร์รัปต์ (interrupt) หรือรับคำสั่ง yield() จากโปรแกรม ผู้เขียนโปรแกรมจะใช้คำสั่ง yield() ในกรณีที่ต้องการให้เชรคอื่นที่คอยใน ready list เข้ามาทำงานแทน ทำให้เชรคที่กำลังทำงานอยู่นั้นหยุดพักการทำงานลงชั่วคราว และจะกลับเข้าทำงานอีกครั้งเมื่อถูกเลือก ready list
  - 4.3 เป็นสถานะ blocked เมื่อเชรคนั้นรับคำสั่ง wait(), suspend() หรือ sleep() จากโปรแกรม หรือทำการติดต่อกับอินพุทเอาต์พุทหรือรอใช้ทรัพยากรของเครื่องที่มีเชรคอื่นกำลังกำลังใช้อยู่
5. เชรคที่อยู่ในสถานะ blocked จะเปลี่ยนสถานะเป็น ready เมื่อได้รับคำสั่ง notify() หรือ resume() จากโปรแกรม หรือเมื่อเสร็จสิ้นการติดต่อกับอินพุทเอาต์พุทหรือได้รับทรัพยากรของระบบที่รอใช้งานอยู่  
สรุปได้ว่า เชรคที่เข้าสู่สถานะ blocked ด้วยคำสั่งข้างล่างนี้ จะกลับสู่สถานะ ready เมื่อมีเหตุการณ์ต่อไปนี้

- wait() : เมื่อมีเชรคอื่นออกคำสั่ง notify()
- suspend() : เมื่อเชรคได้รับคำสั่ง resume()
- sleep(time) : เมื่อครบเวลา time

ปัจจุบันเรามีเครื่องคอมพิวเตอร์สมรรถนะสูงที่มีหน่วยประมวลผลหลาย ๆ ตัว เครื่องคอมพิวเตอร์ประเภทนี้เรียกว่า multiprocessor computer บางคนอาจคิดว่า เมื่อมีหน่วยประมวลผลมากตัวขึ้น โปรแกรมก็น่าจะทำงานเร็วขึ้นตามสัดส่วนของจำนวนหน่วยประมวลผลที่มากขึ้น เช่นเครื่องที่มีหน่วยประมวลผล 4 ตัว ก็ควรที่จะทำงานได้เร็วเป็น 4 เท่าของเครื่องที่มีหน่วยประมวลผลเพียงตัวเดียว แต่ในความเป็นจริงไม่เป็นเช่นนั้น เพราะสถาปัตยกรรมของเครื่องคอมพิวเตอร์นั้นอาจไม่สามารถรองรับการทำงานแบบมีหลาย ๆ หน่วยประมวลผลได้อย่างมีประสิทธิภาพ เนื่องจากอาจมีทรัพยากรของเครื่องไม่เพียงพอต่อการทำงานพร้อม ๆ กัน อีกทั้งระบบปฏิบัติการโดยทั่วไปมักจะไม่สามารถแจกจ่ายโพรเซสหรือเชรค ให้แก่หน่วยประมวลผลหลาย ๆ ตัวให้ทำงานอย่างเต็มที่ได้อย่างคล่องตัวและพร้อม ๆ กัน นอกจากนี้ยังมีปัญหาในระดับของการสร้างโปรแกรม คือ ผู้เขียนโปรแกรมมักไม่ทราบว่าโปรแกรมนี้จะทำงานบนเครื่องที่มีกี่หน่วยประมวลผล จึงไม่สามารถสร้างโปรแกรมให้ใช้ประโยชน์หน่วยประมวลผลที่มีอยู่จริงได้ทั้งหมด แม้ปัจจุบันจะมีคอมไพเลอร์ที่สามารถช่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แตกโปรแกรมทั่วไปออกเป็นโปรแกรมแบบขนาน แต่ก็ทำได้เพียงบางกรณี และมีประสิทธิภาพผู้โปรแกรม ซึ่งถูกออกแบบมาเป็นแบบขนานโดยตรงไม่ได้ สมมติถ้าเราทราบว่าเครื่องที่ใช้ นั้น มีหน่วยประมวลผลกี่ตัว และต้องการให้หน่วยประมวลผลทุกตัวมีงานทำพร้อม ๆ กัน ก็ต้องออกแบบโปรแกรมให้มีเรดเท่ากับจำนวนของหน่วยประมวลผล ซึ่งเป็นเรื่องที่ยุ่ยากมากและไม่สามารถเป็นไปได้ทุกกรณี อีกปัญหาหนึ่งคือระบบปฏิบัติการปัจจุบันจะสามารถทำงานโปรแกรมของผู้ใช้มากกว่าหนึ่งคนในเวลาหนึ่ง นั้นหมายความว่า ในขณะที่โปรแกรมทำงานนั้น อาจมีหน่วยประมวลผลบางตัวทำงานโปรแกรมของคนอื่น ทำให้โปรแกรมใดโปรแกรมหนึ่งไม่สามารถใช้หน่วยประมวลผลได้ทั้งหมดในเวลานั้น ดังนั้น แม้เราจะเพิ่มจำนวนหน่วยประมวลผลเข้าไปในเครื่อง ก็จะทำให้เครื่องทำงานเร็วขึ้นบ้างเท่านั้น แต่จะไม่ได้ทำให้เร็วขึ้นเป็นจำนวนเท่าของหน่วยประมวลผลที่เพิ่มเข้าไป

การที่ระบบปฏิบัติการและภาษาสามารถทำงานแบบมัลติเธรดนั้น ถือว่ามีประโยชน์ช่วยในการใช้ทรัพยากรของเครื่องได้อย่างมีประสิทธิภาพมากขึ้น แต่ต้องอาศัยผู้เขียนโปรแกรมช่วยสร้างโปรแกรมที่ใช้ประโยชน์มัลติเธรดด้วย เนื่องจากการสร้างและควบคุมเธรดในโปรแกรม ไม่ได้เป็นหน้าที่โดยตรงของคอมพิวเตอร์และระบบปฏิบัติการ แต่เป็นหน้าที่ของผู้เขียนโปรแกรมที่จะกำหนดเธรดสำหรับทำงานหน้าที่ต่าง ๆ ที่เกิดขึ้นได้พร้อม ๆ กัน ซึ่งเมื่อทำงานแล้วควรจะทำให้รู้สึกที่เร็วขึ้นกว่าการทำทีละอย่าง แต่ต้องขึ้นกับว่างานนั้นมีความเป็น serializability อยู่มากน้อยเพียงใด คำว่า serializability ในที่นี้หมายถึง การที่งานอย่างหนึ่งสามารถถูกแยกออกเป็นงานย่อย ๆ ได้มากน้อยเท่าไร ถ้าแยกออกได้มากก็ถือว่า serializability อยู่บ่อย แต่ถ้าไม่สามารถแยกออกได้เลยก็ถือว่า serializability 100%

หากเราแยกงานที่มี serializability มากออกเป็นงานย่อย ๆ จะทำให้การทำงานของงานย่อยเหล่านั้นช้าลงกว่าการทำงานอย่างปกติแบบงานเดียวก็ได้ เนื่องจากงานย่อยเหล่านั้นอาจต้องรอผลลัพธ์ของการทำงานอื่นก่อนจะเริ่มทำงานหรือทำงานต่อไปได้ หากแยกงานย่อยเหล่านี้ออกไปทำงานอย่างอิสระจากกัน ก็ไม่สามารถทำงานพร้อมกันได้ แต่กลับจะมีภาระในการส่งค่าติดต่อกันซึ่งอาจจะสูงกว่าการทำงานเพื่อแก้ปัญหา นั้นจริง ๆ ดังนั้นการพยายามแตกโปรแกรมออกเป็นหลายเธรดอย่างไม่สมคณนั้น อาจจะทำให้โปรแกรมทำงานช้าลงกว่าปกติก็ได้

ในการสร้างโปรแกรมสำหรับมัลติเธรด หากเป็นไปได้ควรเลือกให้เธรดแต่ละตัว ทำงานอย่างเป็นอิสระต่อกัน คือควรมีการทำงานให้ขึ้นต่อกันน้อยที่สุดเพื่อไม่ต้องหยุดรอกัน ควรให้มีการส่งค่าไปมาระหว่างเธรดให้น้อยที่สุด และควรให้แต่ละเธรดใช้ทรัพยากรของเครื่องที่แตกต่างกัน เพื่อจะได้ไม่ต้องแย่งกันใช้ ซึ่งจะต้องมีการหยุดรอกัน

## บทที่ 3

### แนวคิดหลักในการพัฒนาโปรแกรม

#### 3.1 บทนำ

เบราว์เซอร์ที่สามารถโปรแกรมได้ (Programmable Browser) คือ เว็บเบราว์เซอร์ที่สามารถทำหน้าที่หลัก ๆ ของเว็บเบราว์เซอร์ทั่ว ๆ ไปได้ คือ แสดงผลไฟล์เอชทีเอ็มแอลและสามารถสั่งงานได้ โดยการเขียนภาษาสคริปต์ขึ้นมาเพื่อใช้ในการสั่งงาน หลักการทำงานของเบราว์เซอร์ที่สามารถโปรแกรมได้ก็เหมือนกับเบราว์เซอร์ทั่ว ๆ ไป แต่เพิ่มความสามารถในการรับคำสั่งจากภาษาสคริปต์เข้าไป

เบราว์เซอร์ที่สามารถโปรแกรมได้ กับ เอเจนต์

การเปรียบเทียบคุณสมบัติของเอเจนต์ กับเบราว์เซอร์ที่สามารถโปรแกรมได้

- เอเจนต์สามารถควบคุมการทำงานของตัวเองได้ สามารถทำกิจกรรมต่าง ๆ ได้อย่างอิสระ โดยไม่ขึ้นกับผู้ใช้หรือสิ่งกระตุ้นภายนอก
- เบราว์เซอร์ที่สามารถโปรแกรมได้ สามารถทำงานได้เองโดยอิสระ หลังจากเขียนสคริปต์สั่งงานเรียบร้อยแล้ว
- เอเจนต์สามารถทำงานได้อย่างต่อเนื่อง
- เบราว์เซอร์ที่สามารถโปรแกรมได้ สามารถใช้ภาษาสคริปต์ในการสั่งงานให้ทำงานต่อเนื่องไปเรื่อย ๆ ได้
- เอเจนต์สามารถสื่อสารกับผู้ใช้หรือเอเจนต์อื่น ๆ ได้
- เบราว์เซอร์ที่สามารถโปรแกรมได้ สามารถสื่อสารกับผู้ใช้ผ่านทางอินเทอร์เน็ตเฟสต่าง ๆ ของเบราว์เซอร์ และผ่านทางภาษาสคริปต์ได้
- เอเจนต์ต้องมีเป้าหมายและกระทำกิจกรรมต่าง ๆ เพื่อบรรลุเป้าหมายนั้น
- เบราว์เซอร์ที่สามารถโปรแกรมได้ ได้รับเป้าหมายในการทำงานผ่านทางภาษาสคริปต์

เว็บเบราว์เซอร์ที่สามารถโปรแกรมได้ ทำให้ผู้ใช้สามารถสั่งงานให้แก่เบราว์เซอร์ได้ โดยผู้ใช้ไม่จำเป็นต้องทำงานนั้น ๆ เอง ลำดับและขั้นตอนการทำงานตามความต้องการของผู้ใช้ เป็นหน้าที่ของเบราว์เซอร์ที่ต้องจัดการ จุดประสงค์สำคัญของการพัฒนาโปรแกรมเว็บเบราว์เซอร์ให้เป็นโปรแกรมเว็บเบราว์เซอร์สามารถถูกโปรแกรมได้นั้น คือ ทำให้ผู้ใช้ได้รับความสะดวกสบายในการรับบริการข้อมูลผ่านทางเครือข่าย ซึ่งรูปแบบในการบริการนั้นมีมากมาย ดังนั้น เพื่อสนับสนุนการรับบริการต่าง ๆ ดังกล่าว เป็นเหตุให้ผู้จัดทำโครงการต้องคำนึงถึงการทำงานและความสามารถพิเศษเพิ่มเติมเป็นส่วนประกอบย่อยของเบราว์เซอร์ที่สามารถโปรแกรมได้ เพื่อการขยายความสามารถของเบราว์เซอร์ ซึ่งจะกล่าวถึงดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2 การจัดการคำสั่งสคริปต์

คำสั่งสคริปต์เป็นส่วนสำคัญส่วนหนึ่งของการทำงานของบราวเซอร์ที่สามารถโปรแกรมได้ โดยบราวเซอร์ที่สามารถโปรแกรมได้นั้นมีแนวทางการทำงานแบ่งได้เป็น 2 แนวทาง คือ

- 1) ผู้ใช้สามารถส่งงานบราวเซอร์ที่สามารถโปรแกรมได้โดยตรง ผ่านอินเทอร์เน็ตที่ทางผู้จัดทำได้กำหนดให้แก่ผู้ใช้ เป็นการตอบสนองการทำงานของผู้ใช้ซึ่งสามารถตอบสนองการให้บริการผ่านเครือข่ายได้เองในขณะนั้น
- 2) ผู้ใช้สามารถส่งงานบราวเซอร์ที่สามารถโปรแกรมได้โดยใช้ภาษาสคริปต์ ตามโครงสร้างที่ผู้จัดทำได้กำหนดขึ้น เป็นการตอบสนองการทำงานของผู้ใช้ที่ไม่สามารถตอบสนองการให้บริการผ่านเครือข่ายได้ในขณะช่วงเวลานั้น

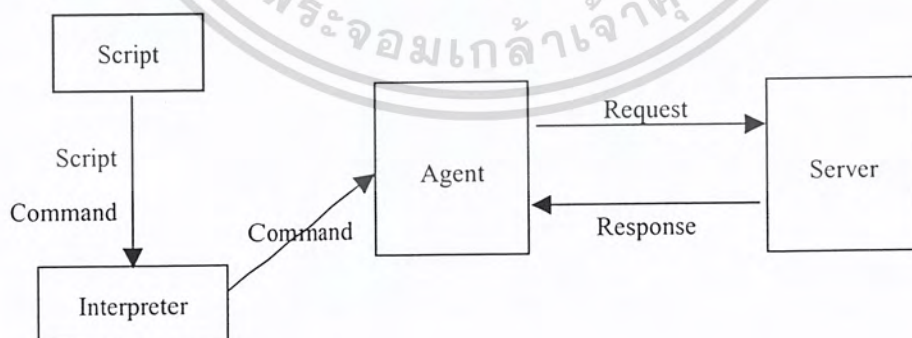
ผู้จัดทำโครงการจัดการการทำงานของคำสั่งสคริปต์ ดังต่อไปนี้

#### 3.2.1 ลักษณะการทำงานของบราวเซอร์ที่สามารถโปรแกรมได้ผ่านภาษาสคริปต์

การทำงานของบราวเซอร์ที่สามารถโปรแกรมได้ผ่านภาษาสคริปต์มีขั้นตอนดังนี้

- 1) การกำหนดคำสั่งสคริปต์ ซึ่งอาจได้มาจากการเขียนคำสั่งสคริปต์โดยผู้ใช้นเิดเตอร์ (editor) ที่สร้างไว้บนเอเจนต์ในขณะนั้น หรืออาจได้มาจากการโหลดคำสั่งสคริปต์ที่ผู้ใช้เคยเก็บไว้มาทำงานได้ ซึ่งคำสั่งสคริปต์ต้องเป็นตามโครงสร้างที่ได้กำหนดไว้
- 2) นำคำสั่งสคริปต์ (Script Command) ที่ได้ ส่งให้อินเทอร์พรีเตอร์ (Interpreter) จัดการ
- 3) อินเทอร์พรีเตอร์ทำการอินเทอร์พรีตคำสั่งสคริปต์ที่ส่งเข้ามา
- 4) มีการสั่งให้เอเจนต์ทำงาน เพื่อการร้องขอและรับบริการผ่านเครือข่าย

ได้อธิบาย ดังรูปที่ 3-1



รูปที่ 3-1 แสดงการทำงานของเอเจนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

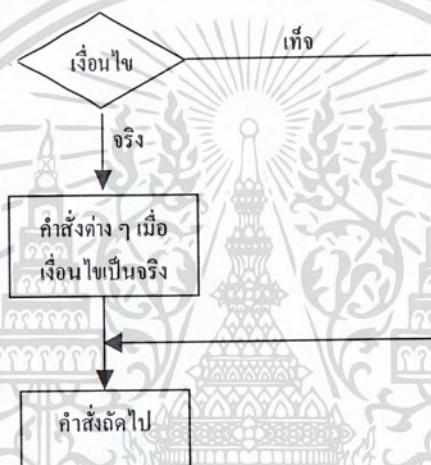
### 3.2.2 ความสามารถของภาษาสคริปต์

ผู้จัดทำโครงการ ได้กำหนดให้ภาษาสคริปต์มีความสามารถดังต่อไปนี้

- 1) ภาษาสคริปต์สามารถทำงานเกี่ยวกับกลุ่มคำสั่งที่ใช้ในการตัดสินใจเลือกเงื่อนไขต่าง ๆ ภายในโปรแกรมเพียงเงื่อนไขเดียวมาทำงาน ซึ่งทางผู้จัดทำโครงการกำหนดให้มีคำสั่ง IF โดยกำหนดให้ทำ 2 แบบ คือ

#### 1.1) แบบเงื่อนไขเดียว (Simple IF)

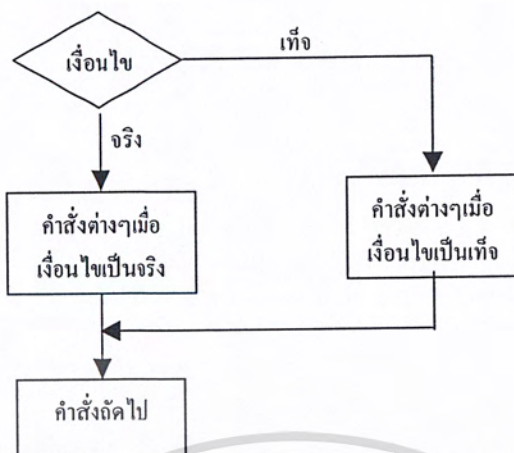
ถ้าตรวจสอบเงื่อนไขแล้วว่าเป็นจริง โปรแกรมจะไปทำงานในส่วนต่อมา แต่ถ้าตรวจสอบเงื่อนไขแล้วเป็นเท็จ จะข้ามส่วนนั้นไป โดยมีการทำงานดังรูปที่ 3-2



รูปที่ 3-2 แสดงการทำงานของภาษาสคริปต์แบบเงื่อนไขเดียว

#### 1.2) แบบ 2 เงื่อนไข (IF...ELSE)

ใช้ในกรณี 2 เงื่อนไข คือ ถ้าเงื่อนไขเป็นจริงให้ทำงานงานหนึ่ง แต่ถ้าเงื่อนไขเป็นเท็จจะต้องไปทำงานอีกงานหนึ่ง โดยมีการทำงาน ดังรูปที่ 3-3

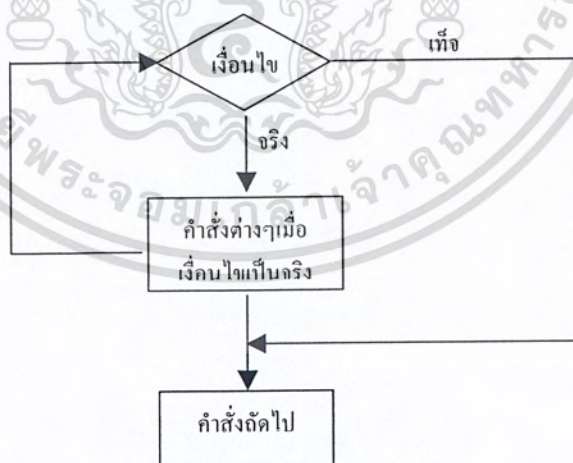


รูปที่ 3-3 แสดงการทำงานของภาษาสคริปต์แบบสองเงื่อนไข

- 2) ภาษาสคริปต์ต้องทำงานเกี่ยวกับกลุ่มคำสั่งที่ใช้ในการวนรอบ (Loop) การทำงานไปเรื่อย ๆ ซึ่งทางผู้จัดทำโครงการกำหนดให้มีคำสั่งการวนรอบ โดยกำหนดให้ทำ 2 แบบ คือ

#### 2.1) คำสั่ง While

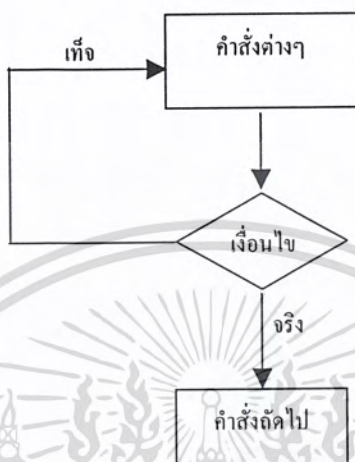
โปรแกรมจะทำงานคำสั่งที่อยู่ภายในคำสั่ง while เมื่อเงื่อนไขที่กำหนดเป็นจริง และจะทำการวนรอบไปเรื่อย ๆ จนกว่าจะตรวจสอบพบว่าเงื่อนไขเป็นเท็จ จะกระโดดออกมาทำงานให้คำสั่งถัดไป โดยมีการทำงานดังรูปที่ 3-4



รูปที่ 3-4 แสดงการทำงานของภาษาสคริปต์แบบดูล while

## 2.2) คำสั่ง Repeat...Until

โปรแกรมจะทำงานคำสั่งที่อยู่ในคำสั่ง Repeat ก่อน 1 ครั้งเสมอ แล้วจึงทำการตรวจสอบเงื่อนไขที่กำหนดไว้(ใน Until) หากเงื่อนไขเป็นเท็จจะยังคงทำการวนรอบไปเรื่อย ๆ แต่เมื่อใดที่เงื่อนไขเป็นจริง จะกระโดดออกมาทำงานให้คำสั่งถัดไป โดยมีการทำงานดังรูปที่ 3-5



รูปที่ 3-5 แสดงการทำงานของภาษาสคริปต์แบบ loop repeat-until

- 3) ภาษาสคริปต์ต้องจัดการคำสั่งที่เป็นความสามารถพิเศษของบราวเซอร์ที่สามารถโปรแกรม เพื่อสั่งการให้เอเจนต์ทำงานตามที่ต้องการได้ ซึ่งแบ่งเป็น 2 ส่วน ดังต่อไปนี้

### 3.1) คำสั่งจัดการบราวเซอร์

#### 3.1.1) open() หรือ open(URL)

เป็นคำสั่งเพื่อสั่งให้เอเจนต์เปิดบราวเซอร์ขึ้น โดยผู้ใช้งานจะกำหนดยูอาร์แอลให้แก่เอเจนต์ เพื่อเปิดบราวเซอร์ที่เชื่อมต่อไปยังยูอาร์แอลที่ผู้ใช้ต้องการได้

#### 3.1.2) close()

เป็นคำสั่งเพื่อสั่งให้เอเจนต์ปิดบราวเซอร์ที่เปิดขึ้นมาแล้ว

### 3.2) คำสั่งจัดการเรื่องซีไอ (CGI)

#### 3.2.1) search(KeyWord)

เป็นคำสั่งเพื่อสั่งให้เอเจนต์ค้นหาเว็บเพจ(web page) ที่ต้องการตามคีย์เวิร์ดที่ผู้ใช้กำหนด แสดงให้แก่ผู้ใช้ได้ ซึ่งเป็นการทำงานแบบ GET

#### 3.2.2) page(Company,Number,Password,Message)

เป็นคำสั่งเพื่อสั่งให้เอเจนต์ส่งข้อความผ่านเครื่องมือสื่อสาร (ในที่นี้หมายถึงเพจเจอร์) ที่ต้องการตามพารามิเตอร์ที่ผู้ใช้กำหนดให้แก่ผู้ใช้ได้ ซึ่งเป็นการทำงานแบบ POST

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4) ภาษาสคริปต์ต้องจัดการคำสั่งที่เกี่ยวข้องกับเวลาได้ เพื่อสั่งการให้เอเจนต์สามารถกำหนดเวลาที่จะเริ่มทำงานได้ โดยเอเจนต์จะทำงานตามที่ต้องการได้ เมื่อถึงเวลาที่ผู้ใช้ได้กำหนดไว้ โดยเขียนเวลาควกับคำสั่งที่จะให้เอเจนต์ หรือหากไม่มีการกำกับเวลาไว้ จะคิดเสมือนว่าให้ทำงาน ณ เวลานั้น
- 5) ภาษาสคริปต์ต้องจัดการคำสั่งสคริปต์ที่เกี่ยวข้องกับตัวแปรได้ เพื่อการทำงานที่ถูกต้อง ซึ่งผู้จัดทำโครงการ ได้กำหนดให้ภาษาสคริปต์นี้จัดการตัวแปรของเลขจำนวนเต็ม (Integer) และตัวแปรของข้อความ (String) ได้
- 6) ภาษาสคริปต์ต้องจัดการคำสั่งสคริปต์ที่เกี่ยวข้องกับการคำนวณได้ เพื่อการทำงานที่ถูกต้อง ซึ่งผู้จัดทำโครงการได้กำหนดให้ภาษาสคริปต์นี้จัดการการคำนวณแบบ บวก และ ลบได้

### 3.3 การทำงานของเอเจนต์

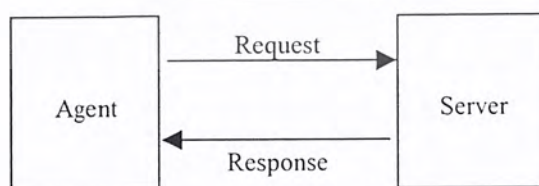
เอเจนต์เป็นส่วนประกอบสำคัญของบราวเซอร์ที่สามารถโปรแกรมได้ ซึ่งเอเจนต์ที่ใช้ในบราวเซอร์ที่สามารถโปรแกรมได้มีชื่อว่า ไทมเมอร์เอเจนต์ (TimerAgent)

#### 3.3.1 ลักษณะการทำงานของเอเจนต์

ลักษณะการทำงานของเอเจนต์ คือ

- 1) เอเจนต์จะทำตัวเสมือนเป็นผู้ร้องขอบริการจากเซิร์ฟเวอร์ เมื่อได้รับคำสั่งจากผู้ใช้ไม่ว่าจะเป็นการสั่งงานผ่านทางยูสเซอร์อินเทอร์เฟซต่าง ๆ หรือการสั่งผ่านคำสั่งสคริปต์ก็ตาม เอเจนต์จะทำการจัดการคำสั่งเหล่านั้นตามอัลกอริทึมที่กำหนด แล้วทำการร้องขอบริการเหล่านั้น
- 2) เมื่อเซิร์ฟเวอร์ หรือผู้ให้บริการ มีการตอบสนองกลับมาที่บราวเซอร์ที่สามารถโปรแกรมได้ จะสั่งการให้บริการมาให้แก่เอเจนต์ หลังจากนั้น เอเจนต์จะเป็นตัวจัดการกับการให้บริการของเซิร์ฟเวอร์ตามอัลกอริทึมที่ระบุ

ได้อธิบาย ดังรูปที่ 3-6



รูปที่ 3-6 แสดงการติดต่อระหว่างเอเจนต์และเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.2 ความสามารถของเอเจนต์

ผู้จัดทำโครงการ ได้กำหนดให้เอเจนต์มีความสามารถดังต่อไปนี้

- 1) เอเจนต์ของบราวเซอร์ที่สามารถโปรแกรมได้นั้น ต้องทำงานแบบเรดโค้ด เพื่อการทำงานที่มีประสิทธิภาพ เนื่องจากบราวเซอร์ที่สามารถโปรแกรมได้นั้น ต้องสามารถทำงานสอดคล้องกับการสั่งงานของผู้ใช้ได้เป็นอย่างดี ซึ่งผู้ใช้อาจสั่งงานโดยผ่านยูสเซอร์อินเทอร์เฟซหรือสั่งงานผ่านคำสั่งสคริปต์ก็ได้ ดังที่ได้กล่าวมาแล้วข้างต้น ซึ่งหากผู้ใช้ สั่งงานโดยใช้คำสั่งสคริปต์ จะมีความเป็นไปได้สูงที่จะเป็นการสั่งงานล่วงหน้าโดยมีการกำหนดเวลาเอาไว้ด้วย หรือสั่งงานให้โปรแกรมทำงานหลาย ๆ อย่าง ซึ่งงานแต่ละงานนั้นอาจใช้เวลาในการทำงานไม่เท่ากัน การกำหนดให้งานแต่ละงาน ทำงานเป็นอิสระต่อกัน ทำให้มีความยืดหยุ่นในการทำงานมากกว่าและมีประสิทธิภาพ ทั้งยังสนับสนุนคุณสมบัติของเอเจนต์ (Autonomous) ได้อีกด้วย
- 2) เอเจนต์ของบราวเซอร์ที่สามารถโปรแกรมได้นั้น ต้องทำงานที่เกี่ยวข้องกับเวลาได้ การนำเวลาเข้ามาเป็นส่วนประกอบหนึ่งในการทำงานจะเพิ่มความสะดวกให้แก่ผู้ใช้อย่างมาก ผู้ใช้สามารถกำหนดเวลาในการทำงานต่าง ๆ ได้ โดยผู้ใช้ไม่จำเป็นต้องรับการตอบสนอง และมอบหน้าที่เหล่านี้ให้แก่เอเจนต์เป็นตัวจัดการ และเก็บข้อมูลที่ได้รับให้แก่ผู้ใช้ เพื่อให้ผู้ใช้สามารถเข้ามาตรวจสอบได้ในภายหลัง และหากผู้ใช้สามารถรับการตอบสนองของเซิร์ฟเวอร์และต้องการใช้บราวเซอร์ที่สามารถโปรแกรมได้โดยตรง ณ เวลานั้น เอเจนต์ก็ต้องสามารถสนับสนุนการทำงานในลักษณะนี้ด้วยเช่นกัน
- 3) เอเจนต์ของบราวเซอร์ที่สามารถโปรแกรมได้นั้น ต้องทำงานตาม que ผู้ใช้กำหนดได้ ซึ่งผู้จัดทำโครงการได้กำหนดการทำงานออกเป็น 2 แบบ คือ

#### 3.1) การทำงานที่เกี่ยวข้องกับบราวเซอร์

3.1.1) เปิดบราวเซอร์ เอเจนต์ต้องสามารถเปิดบราวเซอร์ได้ 2 แบบ คือ

- เปิดบราวเซอร์โดยไม่ได้กำหนดยูอาร์แอลที่ต้องการจะเปิด ซึ่งเมื่อ โปรแกรมทำงาน ผลลัพธ์ที่ได้ คือ หน้าต่างของบราวเซอร์เปิดขึ้น และเชื่อมต่อกับยูอาร์แอลที่ผู้จัดทำโครงการกำหนด
- เปิดโปรแกรมโดยมีการกำหนดยูอาร์แอลโดยผู้ใช้อีก เมื่อ โปรแกรมทำงาน หน้าต่างของบราวเซอร์จะเปิดขึ้น และเชื่อมต่อไปยังยูอาร์แอลที่ผู้ใช้ต้องการ

3.1.2) ปิดบราวเซอร์ โดยผู้จัดทำโครงการกำหนดให้เอเจนต์ปิดบราวเซอร์ที่เปิดขึ้นล่าสุด

#### 3.2) การทำงานที่เกี่ยวข้องกับซีจีไอ(CGI)

3.2.1) ทำงานแบบเก็ท(GET)

ผู้จัดทำโครงการได้กำหนดการทำงานแบบเก็ทให้แก่เอเจนต์ คือ การทำการค้นหาเว็บเพจตามคีย์เวิร์ดที่ผู้ใช้ต้องการ แล้วเก็บเว็บเพจนั้นไว้ให้แก่ผู้ใช้ในรูปแบบของไฟล์นามสกุลเอชทีเอ็มแอล (HTML Files) ซึ่งเซิร์ฟเวอร์ที่เอเจนต์จะติดต่อนั้น ผู้จัดทำโครงการได้กำหนดไว้ให้ รวมทั้งได้ศึกษารูปแบบของการส่งซีจีไอของเซิร์ฟเวอร์นี้เรียบร้อยแล้ว เอ

เจเนตเพียงแต่แปลงคีย์เวิร์ดที่ได้รับให้อยู่ในรูปของยูอาร์แอลตามเงื่อนไขที่กำหนด เพื่อส่ง  
ไปให้เซิร์ฟเวอร์

### 3.2.2) ทำงานแบบโพสต์ (POST)

ผู้จัดทำโครงการได้กำหนดการทำงานแบบโพสต์ให้แก่เจเนต คือ การส่งข้อความ  
เข้าสู่เครื่องเพจเจอร์ โดยผู้จัดทำโครงการได้กำหนดเซิร์ฟเวอร์และรูปแบบการส่งซีจีไอไว้  
แล้วเช่นกัน

## 3.4 การส่งและรับข่าวของบราวเซอร์ที่สามารถโปรแกรมได้ผ่าน NNTP

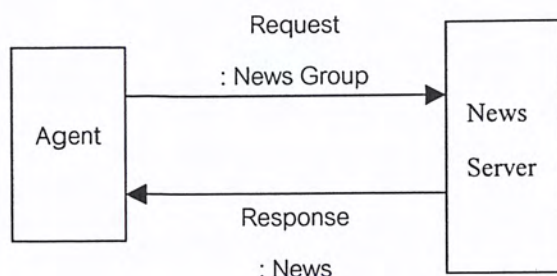
การทำงานต่อไปที่จะกล่าวถึงของบราวเซอร์ที่สามารถโปรแกรมได้ คือ การส่งและรับข่าวสารโดย  
ผ่าน เน็ตเวิร์กนิวส์ทรานส์เฟอร์ โพรโตคอล หรือ เอ็นเอ็นทีพี (Network News Transfer Protocol : NNTP) ซึ่ง  
เป็นโพรโตคอลที่ใช้จัดการข้อความหรือข่าวสารต่าง ๆ ที่ถูกโพสต์ (post) ลงบนยูสเน็ต (Usenet)

### 3.4.1 ลักษณะการรับและส่งข่าวสารของบราวเซอร์ที่สามารถโปรแกรมได้ผ่านเอ็นเอ็นทีพี

การรับและการส่งข่าวของบราวเซอร์ที่สามารถโปรแกรมได้ผ่านเอ็นเอ็นทีพี เป็นดังนี้

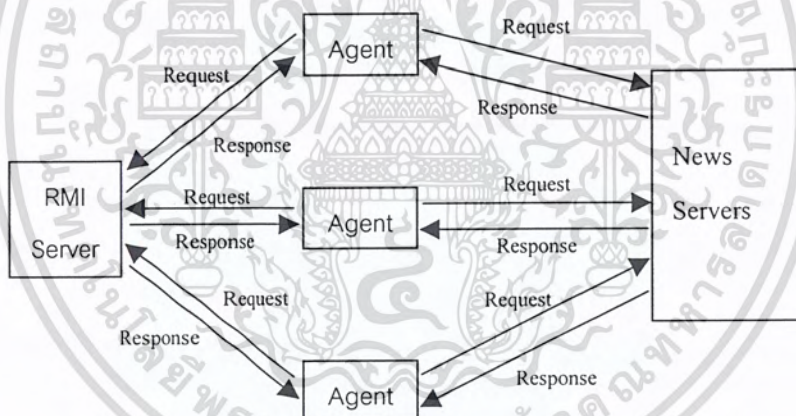
- 1) เจเนตของบราวเซอร์ที่สามารถโปรแกรมได้ทำตัวเสมือนไคลเอ็นต์ที่ร้องขอบริการจาก  
เซิร์ฟเวอร์ ซึ่งการรับส่งข่าวผ่านโพรโตคอลชนิดนี้ จะมีนิวส์เซิร์ฟเวอร์ (News Server) เป็นผู้  
ให้บริการ ส่งข่าวสารให้แก่ไคลเอ็นต์ เพื่อให้ได้มาซึ่งข่าวสารที่ต้องการ อันดับแรกผู้ใช้ต้องทำ  
การเชื่อมต่อเข้ากับนิวส์เซิร์ฟเวอร์ นิวส์เซิร์ฟเวอร์ที่ให้บริการอาจมีได้หลายนิวส์เซิร์ฟเวอร์แล้ว  
แต่ความต้องการของผู้ใช้ เมื่อสามารถเชื่อมต่อกับนิวส์เซิร์ฟเวอร์ได้แล้ว ผู้ใช้จำเป็นต้องระบุนิวส์กรู  
ป (News Group) ให้แก่โปรแกรมด้วย เพื่อเป็นการบอกให้นิวส์เซิร์ฟเวอร์คัดเลือกข่าวสารหรือ  
ข้อความต่าง ๆ ตามหัวข้อของนิวส์กรู๊ปที่กำหนด ได้อย่างถูกต้อง
- 2) เมื่อนิวส์เซิร์ฟเวอร์ได้รับนิวส์กรู๊ปและได้ทำการค้นหาข่าวสารเรียบร้อยแล้ว จะส่งข้อมูลข่าวสาร  
ที่ได้กลับมายังไคลเอ็นต์ ซึ่งในที่นี้หมายถึงเจเนตของบราวเซอร์ที่สามารถโปรแกรมได้นั่นเอง
- 3) เมื่อเจเนตได้รับข่าวสารจากนิวส์เซิร์ฟเวอร์แล้ว เจเนตจะเป็นตัวจัดการข้อมูลเหล่านั้น เพื่อ  
ความสะดวกแก่การตรวจสอบข้อมูลให้แก่ผู้ใช้

อธิบายลักษณะการทำงานได้ ดังรูปที่ 3-7



รูปที่ 3-7 แสดงการติดต่อระหว่างเอเจนต์และนิวส์เซิร์ฟเวอร์

นอกจากนี้ ผู้จัดทำโครงการ ได้เพิ่มความสามารถพิเศษให้แก่บราวเซอร์ที่สามารถโปรแกรมได้เมื่อเอเจนต์ได้รับข่าวสารมาจากนิวส์เซิร์ฟเวอร์แล้ว คือ เอเจนต์สามารถส่งข่าวสารที่ได้มาเหล่านั้นให้แก่เอเจนต์ตัวอื่น ๆ ที่ทำงานอยู่ในขณะนั้นได้ โดยอาศัยหลักการทำงานของรีโมตเมธอดอินโวก์ชัน(อาร์เอ็มไอ : RMI) ซึ่งต้องมีเซิร์ฟเวอร์ที่สามารถจัดการการทำงานเหล่านี้ได้ ซึ่งอธิบายได้ดังรูปที่ 3-8



รูปที่ 3-8 แสดงการประยุกต์ใช้รีโมตเมธอดอินโวก์ชันกับเอเจนต์

### 3.4.2 ความสามารถของเอเจนต์ที่สนับสนุนการส่งและรับข้อมูลผ่านเอ็นเอ็นทีพี

ความสามารถของเอเจนต์ที่จะกล่าวถึงในที่นี้ จะหมายรวมถึงความสามารถของเอเจนต์ในการจัดการข่าวสารให้แก่เอเจนต์ตัวอื่น ๆ โดยใช้หลักการของรีโมตเมธอดอินโวก์ชันด้วย เอเจนต์ที่สนับสนุนการทำงานทั้งหมดนี้ ต้องมีความสามารถดังนี้

- 1) เอเจนต์ต้องสามารถร้องขอการให้บริการจากนิวส์เซิร์ฟเวอร์ได้อย่างถูกต้อง นั่นคือ เอเจนต์ต้องมีการจัดการหรือมีการติดต่อกับผู้ใช้เพื่อให้ได้มาซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ชื่อของนิวส์เซิร์ฟเวอร์ เพื่อทำการเชื่อมต่อและขอใช้บริการจากนิวส์เซิร์ฟเวอร์ที่ต้องการได้ ซึ่งผู้จัดทำโครงการกำหนดให้เอเจนต์เชื่อมต่อกับนิวส์เซิร์ฟเวอร์ผ่านซ็อกเก็ต และใช้พอร์ต 119 ซึ่งเป็นไปตามเงื่อนไขของโปรโตคอล
- ชื่อของนิวส์กรุป เพื่อเป็นการกำหนดหัวเรื่องหรือประเภทของข่าวที่ผู้ใช้ต้องการ เนื้อหาและรายละเอียดของข่าวสารนั้นจะสอดคล้องกับหัวเรื่องที่กำหนด
  - 2) เอเจนต์ต้องสามารถทำการสื่อสารผ่านโปรโตคอลได้อย่างถูกต้อง และเมื่อได้รับข่าวสารมาแล้ว ต้องทำการจัดการข่าวสารนั้น ให้อยู่ในรูปแบบที่ง่ายต่อการอ่านและการตรวจสอบโดยผู้ใช้ ซึ่งข่าวสารที่ได้นั้นอาจมีได้มากกว่า 1 ข่าวสาร ผู้จัดทำโครงการจึงได้กำหนดการแสดงผลลัพธ์เป็น 2 ส่วน ดังนี้
- รายการแสดงหัวข้อข่าว เมื่อข่าวสารที่ได้รับมีมากกว่า 1 ข่าวสาร สร้างความสะดวกให้แก่ผู้ใช้ เพื่อให้สามารถเลือกดูรายละเอียดข่าวเฉพาะหัวข้อข่าวที่ต้องการได้
- ส่วนแสดงรายละเอียดของข่าว ซึ่งต้องสอดคล้องกับหัวข้อข่าวตามรายการที่ผู้ใช้เลือก และสามารถเปลี่ยนแปลงตามการเลือกของหัวข้อข่าวได้
  - 3) เอเจนต์สามารถส่งข่าวสารที่ได้รับให้แก่เอเจนต์ตัวอื่นได้ แต่ผู้จัดทำโครงการได้กำหนดให้มีการระบุเอเจนต์อื่นที่สามารถรับข่าวสารได้ โดยสามารถระบุเอเจนต์ปลายทางผ่านไฟล์ที่เก็บข้อมูลของเอเจนต์ต้นทางแต่ละตัว ผู้จัดทำโครงการเรียกไฟล์ที่เก็บข้อมูลของเอเจนต์ต้นทางนี้ว่า ยูสเซอร์โปรไฟล์ (UserProfile) ซึ่งรายละเอียดของไฟล์นี้ ประกอบด้วย
    - ชื่อของเอเจนต์ต้นทาง
    - นิวส์กรุปที่เอเจนต์ต้นทางสนใจ
    - ชื่อของผู้ที่ต้องการจะส่งข่าวสาร ไปให้ หรือเอเจนต์ปลายทางนั่นเอง
  - 4) แนวคิดในการส่งข่าวสารให้แก่เอเจนต์ปลายทางคือ
    - 4.1) เอเจนต์ต้นทางต้องแจ้งความประสงค์ในการส่งข่าวสารให้แก่เอเจนต์ตัวอื่น เพื่อการเปิดการเชื่อมต่อไปสู่นิวส์เซิร์ฟเวอร์ที่ให้บริการ โดยหลักการรีโมตเมธอดอินโวนเคชัน นั่นคือ ผู้ใช้สามารถเลือกได้ว่า จะส่งข่าวสารให้แก่เอเจนต์ตัวอื่นหรือไม่
    - 4.2) กรณีที่ผู้ใช้ต้องการส่งข่าวสารให้แก่เอเจนต์ตัวอื่น เอเจนต์ต้องทำการตรวจสอบยูสเซอร์โปรไฟล์ เพื่อทำการลงทะเบียนตนเอง (ชื่อของเอเจนต์ต้นทาง) ให้แก่เซิร์ฟเวอร์ผู้ให้บริการ
    - 4.3) เมื่อเอเจนต์ต้นทางได้รับข่าวสาร จะทำการส่งข่าวสารที่ได้นั้นให้แก่เอเจนต์ตัวอื่น ซึ่งเอเจนต์นั้น ระบุจากชื่อของเอเจนต์ปลายทางที่กำหนดในยูสเซอร์โปรไฟล์ โดยเอเจนต์ต้องทำการตรวจสอบกับเซิร์ฟเวอร์ว่าเอเจนต์นั้นทำงานอยู่ ณ เวลานั้นหรือไม่ จึงจะสามารถส่งข่าวสารให้ได้ นอกจากนี้ ต้องมีการตรวจสอบถึงนิวส์กรุปที่เอเจนต์ปลายทางสนใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วย หากเอเจนต์ปลายทางมีความสนใจตรงกับหัวข้อข่าวที่เอเจนต์ต้นทางได้รับมา เอเจนต์ต้นทางจึงจะส่งข่าวสารให้แก่เอเจนต์ปลายทาง

4.4) เอเจนต์ปลายทางจะได้รับข่าวสารในรูปแบบของเท็กซ์ไฟล์ (Text File) เมื่อรับข่าวสารเรียบร้อยแล้ว จะมีหน้าต่างบอกให้เอเจนต์ปลายทางรู้ว่าเมื่อเอเจนต์อื่นส่งข่าวสารมาให้ และสามารถเข้าไปดูได้

5) ความสามารถของเซิร์ฟเวอร์ที่จัดการ มีดังนี้

- 5.1) สามารถเก็บการลงทะเบียนของเอเจนต์ที่เชื่อมต่อได้
- 5.2) สามารถแก้ไขการลงทะเบียนได้เมื่อเอเจนต์ได้ยกเลิกการติดต่อกับเซิร์ฟเวอร์
- 5.3) สามารถส่งข่าวสารให้แก่เอเจนต์ที่เชื่อมต่อเพื่อรับบริการได้

### 3.5 การทำงานของบราวเซอร์ที่สามารถโปรแกรมได้ผ่านโปรแกรมเทลเน็ต (Telnet)

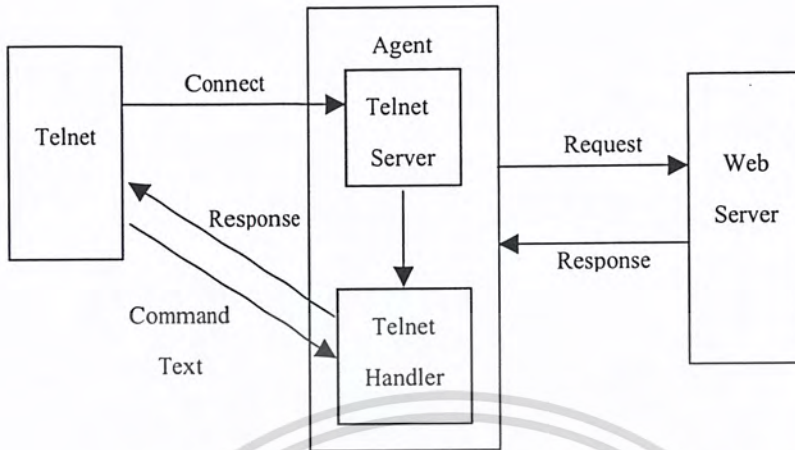
เพื่อเพิ่มความสามารถและความยืดหยุ่นในการใช้งานของบราวเซอร์ที่สามารถโปรแกรมได้ ผู้จัดทำโครงการจัดทำให้บราวเซอร์ที่สามารถโปรแกรมได้ สามารถถูกสั่งการได้จากโปรแกรมระยะไกล ซึ่งทางผู้จัดทำโครงการได้กำหนดให้มีการสั่งการระยะไกลผ่าน โปรแกรมเทลเน็ต

#### 3.5.1 ลักษณะการทำงานของบราวเซอร์ที่สามารถโปรแกรมได้ผ่านโปรแกรมเทลเน็ต

การทำงานของบราวเซอร์ที่สามารถโปรแกรมได้มีขั้นตอน ดังต่อไปนี้

- 1) ผู้ใช้สามารถสั่งงานบราวเซอร์จากระยะไกลผ่านโปรแกรมเทลเน็ตได้ โดยผู้ใช้ต้องทำการเชื่อมต่อกับโฮสต์ของบราวเซอร์ที่สามารถโปรแกรมได้ตามที่ผู้ใช้ต้องการ รวมทั้งผู้ใช้ต้องระบุพอร์ตสำหรับการเชื่อมต่อให้ถูกต้อง
- 2) เมื่อโปรแกรมเทลเน็ตสามารถเชื่อมต่อกับเอเจนต์ได้แล้ว โปรแกรมเทลเน็ตสามารถสั่งงานให้กับเอเจนต์ได้
- 3) ในการเปิดบราวเซอร์ที่สามารถโปรแกรมได้ จะทำการสร้างเซิร์ฟเวอร์เฉพาะแต่ละเครื่องทันที เมื่อผู้ใช้สามารถติดต่อกับโฮสต์ซึ่งหมายถึงบราวเซอร์ที่สามารถโปรแกรมได้สำเร็จ เซิร์ฟเวอร์จะทำการสร้างตัวจัดการการเชื่อมต่อ (Telnet Handler) ระหว่างโปรแกรมเทลเน็ต และบราวเซอร์ที่สามารถโปรแกรมได้
- 4) เอเจนต์ทำงานตามคำสั่งที่ได้รับในรูปแบบต่าง ๆ กัน โดยทำหน้าที่เสมือนเป็นโคลเอเจนต์ติดต่อกับเว็บเซิร์ฟเวอร์

อธิบาย ดังรูปที่ 3-9



รูปที่ 3-9 แสดงการทำงานของบราวเซอร์ผ่านโปรแกรมเทลเน็ต

### 3.5.2 ความสามารถของโปรแกรมเทลเน็ต

โปรแกรมเทลเน็ตที่สนับสนุนการทำงานของบราวเซอร์ที่สามารถโปรแกรมได้ ต้องมีคุณสมบัติดังต่อไปนี้

- 1) โปรแกรมเทลเน็ตต้องสามารถติดต่อกับบราวเซอร์ที่สามารถโปรแกรมได้อย่างถูกต้องผ่านเซิร์ฟเวอร์ซึ่งจะทำงานทุกครั้งเมื่อมีการเปิดบราวเซอร์ที่สามารถโปรแกรมได้ ผู้จัดทำโครงการเรียกเซิร์ฟเวอร์นี้ว่า เทลเน็ตเซิร์ฟเวอร์ (Telnet Server) การเชื่อมต่อระหว่างโปรแกรมเทลเน็ตกับบราวเซอร์ที่สามารถโปรแกรมได้ ต้องเชื่อมต่อและส่งข้อมูลแบบซ็อกเก็ต สิ่งที่ทำเป็นในการระบวการเชื่อมต่อกับบราวเซอร์ได้คือ
  - ชื่อโฮสต์ของบราวเซอร์ที่สามารถโปรแกรมได้ เพื่อส่งคำสั่งจากผู้ใช้ที่อยู่ทั่วโลกให้แก่บราวเซอร์ได้
  - ระบุพอร์ตในการเชื่อมต่อกับโฮสต์ได้อย่างถูกต้อง ในที่นี้ ผู้จัดทำโครงการได้กำหนดพอร์ต 6666 เป็นพอร์ตที่ใช้ในการเชื่อมต่อระหว่างโปรแกรมเทลเน็ต และบราวเซอร์ที่สามารถโปรแกรมได้
- 2) เมื่อโปรแกรมเทลเน็ตสามารถเชื่อมต่อกับบราวเซอร์ได้แล้ว เทลเน็ตเซิร์ฟเวอร์จะทำการสร้างตัวจัดการการเชื่อมต่อกับโปรแกรมเทลเน็ตทั้งหมด เรียกว่า เทลเน็ต แฮนเดอร์ (Telnet Handler) และหลังจากนั้น โปรแกรมเทลเน็ตจะติดต่อกับบราวเซอร์ที่สามารถโปรแกรมได้ผ่านเทลเน็ตแฮนเดอร์ทั้งหมด
- 3) โปรแกรมเทลเน็ตสนับสนุนความสามารถในการรักษาความปลอดภัย นั่นคือ การสั่งงานบราวเซอร์จากระยะไกล เมื่อสามารถเชื่อมต่อกับโฮสต์ (บราวเซอร์ที่สามารถโปรแกรมได้) ได้แล้ว ต้องมีการตรวจสอบรหัสสำหรับการเข้าใช้งานสำหรับผู้ใช้นั้น ๆ เพื่อป้องกันการใช้งานอย่างไม่เหมาะสมจากผู้ที่ไม่ได้รับอนุญาต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4) โปรแกรมเทลเน็ตต้องมีคำสั่งสนับสนุนการสั่งงานให้แก่บราวเซอร์ ซึ่งคำสั่งที่ใช้จะเป็นการสั่งโดยใช้คำสั่งภาษาสคริปต์ ซึ่งมีรายละเอียดดังกล่าวข้างต้น ผู้จัดทำโครงการได้กำหนดคำสั่งเพื่อใช้การสั่งงานบราวเซอร์บน โปรแกรมเทลเน็ตเป็น 3 คำสั่งดังนี้
- 4.1) คำสั่ง `editscript` เมื่อสั่งคำสั่งนี้ผ่านโปรแกรมเทลเน็ตแล้ว ผู้ใช้สามารถพิมพ์คำสั่งสคริปต์ลงบนโปรแกรมได้เลย ซึ่งคำสั่งสคริปต์นั้นต้องมีอัลกอริทึมตามภาษาสคริปต์ที่กำหนดไว้ ดังรายละเอียดข้างต้น
  - 4.2) คำสั่ง `runscript` เมื่อสั่งคำสั่งนี้ผ่านโปรแกรมเทลเน็ต ผู้ใช้ต้องพิมพ์ชื่อไฟล์ซึ่งเป็นไฟล์สคริปต์ที่ต้องการสั่งให้บราวเซอร์ทำงานลงไป ซึ่งไฟล์สคริปต์ที่พิมพ์ลงไปนั้น จะเป็นไฟล์ในฝั่งของโฮสต์ หรือบราวเซอร์ที่สามารถโปรแกรมได้นั้นเอง
  - 4.3) คำสั่ง `exit` ใช้สำหรับออกจากโปรแกรมเทลเน็ต
- 5) ความสามารถของเทลเน็ตแฮนเดอร์
- 5.1) สามารถทำเอคโค (Echo) ได้ เพื่อแสดงการตอบสนองต่าง ๆ บนโปรแกรมเทลเน็ตได้
  - 5.2) สามารถแปลงคำสั่งที่ได้จากโปรแกรมเทลเน็ต (Command Text) ให้อยู่ในรูปแบบที่ตัวจัดการในการทำงานคำสั่งสคริปต์บนบราวเซอร์สามารถเข้าใจได้ เพื่อนำไปสั่งงานเอเจนต์ต่อไป
  - 5.3) สามารถจัดการเรื่องการรักษาความปลอดภัยจากผู้ที่ไม่ได้รับอนุญาตให้สั่งงานบราวเซอร์จากระยะไกลได้ โดยตรวจสอบชื่อและรหัสของผู้ที่จะใช้งานบราวเซอร์ในระยะไกลได้อย่างถูกต้อง

## บทที่ 4

### การพัฒนาโปรแกรม

#### 4.1 บทนำ

จากบทที่แล้ว ได้กล่าวถึงแนวคิดหลักในการพัฒนาโปรแกรมบราวเซอร์ที่สามารถโปรแกรมได้ (Programmable Browser) ซึ่งเป็นขั้นตอนในการออกแบบโครงสร้างการทำงานของโปรแกรมในส่วนต่าง ๆ รวมทั้งพิจารณาและกำหนดความสามารถที่สำคัญให้แก่โปรแกรม เพื่อการเขียนโปรแกรมได้อย่างมีประสิทธิภาพ รวมทั้งมีขอบเขตที่เป็นไปได้จริง

บราวเซอร์ที่สามารถโปรแกรมได้แบ่งการทำงานออกเป็น 4 ส่วนหลัก ดังนี้

1. ส่วนของการจัดการคำสั่งสคริปต์
2. ส่วนของการทำงานของเอเจนต์
3. ส่วนของการส่งและรับข่าวสารผ่าน NNTP
4. ส่วนของการสั่งงานโปรแกรมในระยะไกลผ่านโปรแกรมเทลเน็ต

แนวคิดการทำงานของบราวเซอร์ที่สามารถโปรแกรมได้ก็แบ่งออกได้ตามการทำงานหลักทั้ง 4 ส่วน ซึ่งสามารถสรุปสาระสำคัญได้ดังนี้ ส่วนของการจัดการคำสั่งสคริปต์ ต้องพิจารณาความสามารถพื้นฐานที่คำสั่งสคริปต์ควรทำได้ เช่น การตรวจสอบเงื่อนไข การวนรอบ การสนับสนุนการสั่งงานที่บราวเซอร์ควรทำได้ สนับสนุนการคำนวณและการใช้ตัวแปรได้ เป็นต้น ซึ่งความสามารถของภาษาที่ผู้จัดทำโครงการได้กำหนดขอบเขตไว้ยังไม่ครอบคลุมมากนัก ส่วนของการทำงานของเอเจนต์ ต้องสามารถตั้งเวลา และทำงานตามความสามารถที่ควรทำได้ เช่น สามารถเปิดและปิดบราวเซอร์ได้ ส่วนของการส่งและรับข่าวสารผ่าน NNTP ต้องสามารถนำข่าวมาจากนิวส์เซิร์ฟเวอร์ได้ และแลกเปลี่ยนข้อมูลกับเอเจนต์ตัวอื่นได้ และส่วนสุดท้าย ก็คือส่วนของการสั่งงานโปรแกรมระยะไกลผ่านโปรแกรมเทลเน็ต จะต้องทำงานได้ตามคำสั่งจากผู้ใช้ เช่นเกี่ยวกับการสั่งงานไปยังโปรแกรมโดยตรง

การทำงานของบราวเซอร์ที่สามารถโปรแกรมได้ทั้ง 4 ส่วนนั้น ต่างมีความสัมพันธ์กันทั้งสิ้น คือ การทำงานทุกส่วนจะสามารถทำงานได้ต้องผ่านเอเจนต์เท่านั้น โดยส่วนต่าง ๆ จะเป็นเสมือนตัวแปลงการสั่งงานจากผู้ใช้ ให้อยู่ในรูปแบบการทำงานที่เอเจนต์สามารถเข้าใจได้ แล้วเรียกเอเจนต์ให้มาทำคำสั่งเหล่านั้นต่อ เมื่อเอเจนต์ทำงาน ผู้จัดทำโครงการจะมองเอเจนต์ว่าเป็นเสมือนไคลเอ็นต์หรือผู้ร้องขอบริการผ่านเครือข่ายทำการร้องขอบริการจากผู้ให้บริการหรือเซิร์ฟเวอร์ การทำงานหลักของบราวเซอร์ที่สามารถโปรแกรมได้ทั้ง 4 ส่วนนั้น ต่างก็มีหน้าที่และแนวทางในการทำงานที่แตกต่างกันไป ย่อมมีผลต่อการอิมพลีเมนต์โปรแกรมในส่วนต่าง ๆ เหล่านี้ด้วย ซึ่งผู้จัดทำโครงการได้ทำการอิมพลีเมนต์การทำงานในส่วนต่าง ๆ แยกออกจากกัน นั่นคือทำการอิมพลีเมนต์ในส่วนย่อยก่อน เมื่ออิมพลีเมนต์ส่วนย่อยเสร็จแล้ว ผู้จัดทำโครงการต้องทำการทดสอบโปรแกรมที่อิมพลีเมนต์แล้ว หากผลที่ได้จากการทดสอบไม่ถูกต้อง ผู้จัดทำโครงการต้องทำการแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมนั้นให้ทำงานได้อย่างสมบูรณ์ตามที่กำหนด เมื่อโปรแกรมทำงานได้ถูกต้องแล้ว จึงนำมารวมกับโปรแกรมจัดการการทำงานย่อยในส่วนอื่นที่ผ่านขั้นตอนดังกล่าวแล้วเช่นกัน แล้วทำการทดสอบตามกระบวนการเดิม ทำเช่นนี้จนโปรแกรมทั้งหมดทำงานได้อย่างสมบูรณ์

## 4.2 การจัดการคำสั่งสคริปต์

ส่วนประกอบสำคัญของการอิมพลิเมนต์การเขียนโปรแกรม มีดังต่อไปนี้

### 4.2.1 ความสามารถในการจัดการตัวแปรได้

เป็นความสามารถที่ภาษาสคริปต์สามารถกำหนดค่าให้แก่ตัวแปรได้ ตามชนิดของตัวแปรที่กำหนด เช่น การกำหนดตัวแปรชื่อ  $x$  ชนิดเลขจำนวนเต็ม หรือ อินทิเจอร์ (Integer) ให้มีค่าได้ แนวทางการจัดการตัวแปร มี 2 ทาง คือ

- กำหนดค่าให้แก่ตัวแปร โดยตรงเช่น กำหนดตัวแปร  $x$  ให้มีค่าเท่ากับ 2 เป็นต้น
- สามารถนำตัวแปรมาคำนวณทางคณิตศาสตร์ หรือตรวจสอบค่าได้ เช่น สามารถตรวจสอบได้ว่า ตัวแปร  $x$  มีค่ามากกว่า 4 หรือไม่ เป็นต้น

ผู้จัดทำโครงการได้ทำการอิมพลิเมนต์โปรแกรม โดยกำหนดตัวแปรที่ใช้ในการเก็บข้อมูลของตัวแปรในภาษาสคริปต์ ในที่นี้ ผู้จัดทำโครงการได้กำหนดตัวแปรชนิดแฮชเทเบิล (Hashtable) ขึ้น โดยให้ชื่อว่า `variableInt` และ `variableString` เพื่อใช้ในการเก็บค่าของตัวแปรของภาษาสคริปต์ที่เป็นชนิดอินทิเจอร์ และ สตริง (String) ตามลำดับ

`variableInt` มีคีย์เป็น ชื่อตัวแปร ชนิดสตริง และมีค่าเป็น ค่าของตัวแปร ชนิดอินทิเจอร์

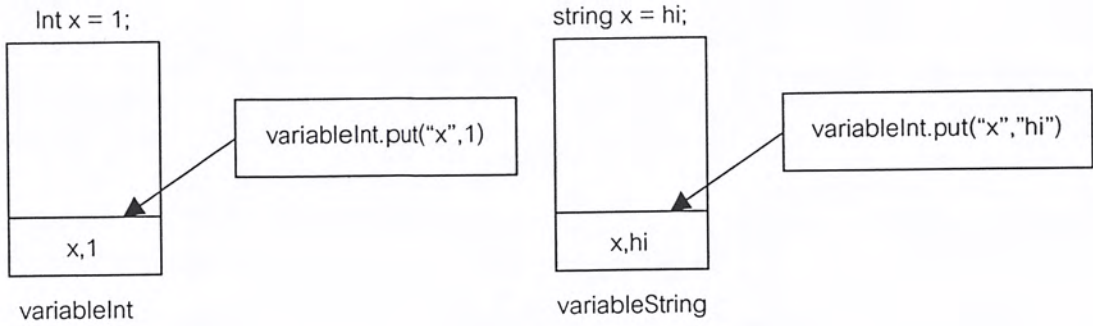
`variableString` มีคีย์เป็น ชื่อตัวแปร ชนิดสตริง และมีค่าเป็น ค่าของตัวแปร ชนิดสตริง

จากคุณสมบัติของแฮชเทเบิล ทำให้ผู้จัดทำโครงการสามารถจัดการตัวแปรของภาษาสคริปต์ได้ ดังนี้

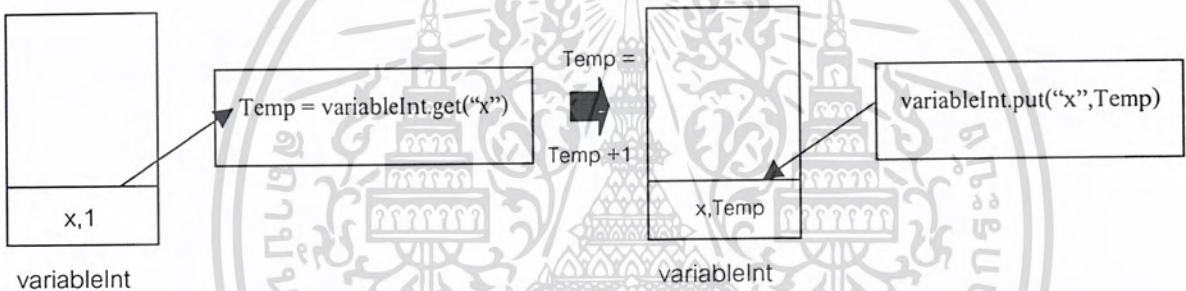
- เมื่อต้องการป้อนค่าหรือต้องการเปลี่ยนแปลงค่าให้กับตัวแปร ผู้จัดทำโครงการสามารถจัดการได้โดยใช้เมธอด `put(Object key, Object value)` เพื่อเป็นการกำหนดว่า ตัวแปรคีย์ มีค่าเป็นแวลู (value)
- เมื่อต้องการนำค่าจากตัวแปรในภาษาสคริปต์มาทำการคำนวณ หรือตรวจสอบทางตรรกศาสตร์ ผู้จัดทำโครงการสามารถจัดการได้โดยใช้เมธอด `get(Object key)` เพื่อเอาค่าของตัวแปรคีย์ออกมา

ลักษณะของการใช้เมธอด `put` และ `get` เพื่อประโยชน์ในการจัดการตัวแปรในภาษาสคริปต์ แสดงด้วย

รูปที่ 4-1 และ รูปที่ 4-2



รูปที่ 4-1 แสดงการเก็บค่าของตัวแปรแบบ Integer และ String



รูปที่ 4-2 แสดงการแก้ไขค่าของตัวแปร

4.2.2 ความสามารถในการทำคำสั่ง IF ได้ ซึ่งแบ่งเป็น 2 แบบ คือ

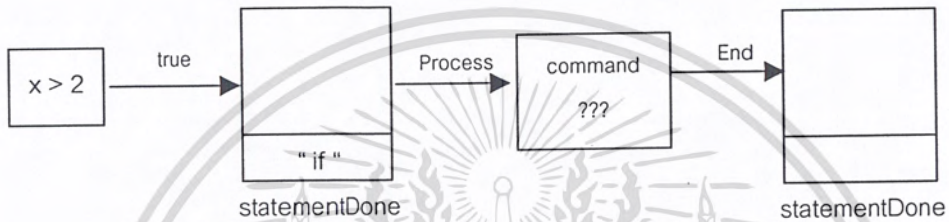
- IF statement

การอิมพลิเมนต์โปรแกรมที่ทำงานในการจัดการคำสั่งสคริปต์ เพื่อทำคำสั่ง IF นั้น ต้องกำหนดตัวแปรที่สำคัญ คือ ตัวแปร statementDone ชนิดเวกเตอร์ ซึ่งเป็นตัวแปรที่ใช้เก็บคำสั่ง if, else, while และ repeat เมื่อเงื่อนไขที่กำหนดถูกต้อง และนอกจากนี้ยังมีตัวแปรที่เกี่ยวข้องอีก 1 ตัว คือ statementNotDone ซึ่งเป็นชนิดเวกเตอร์เช่นเดียวกับ statementDone แต่จะใช้ในกรณีที่เงื่อนไขเป็นเท็จ ดังนั้นการทำงานของคำสั่ง if แบ่งเป็น 2 กรณี ในกรณีแรกคือ เมื่อ เงื่อนไขถูกต้อง โดยมีขั้นตอนในการทำงาน ดังนี้

- 1) เมื่อพบคำสั่ง if และตรวจสอบเงื่อนไขพบว่าจริงแล้ว จะเก็บคำสั่ง if ไว้ใน statementDone
- 2) ทำชุดคำสั่ง เมื่อเงื่อนไขถูกต้อง
- 3) เมื่อหมดชุดคำสั่ง โปรแกรมจะกำจัดค่าที่เก็บไว้ที่ statementDone ซึ่งเป็นค่าล่าสุดออกไป

จากขั้นตอนข้างต้น สามารถอธิบายได้ดังรูปที่ 4-3 ซึ่งเป็นการแสดงการทำงานของโปรแกรม ที่เขียนโดยคำสั่งสคริปต์ ดังนี้ (กำหนดให้  $x = 4$ )

```
if (x>2) then
begin
    ??? (ชุดคำสั่ง)
end;
```

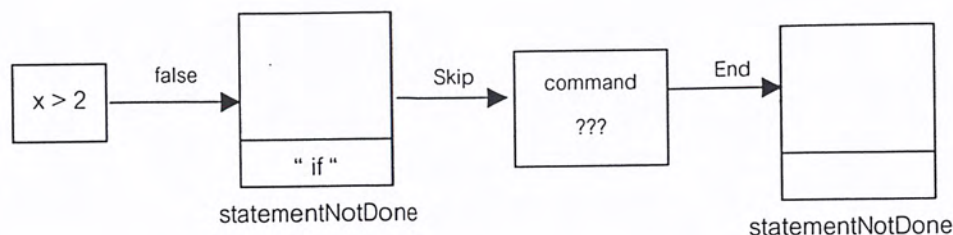


รูปที่ 4-3 แสดงการทำงานของคำสั่ง if เมื่อเงื่อนไขเป็นจริง

ในกรณีที่ 2 คือ เมื่อเงื่อนไขในคำสั่ง if เป็นเท็จ มีขั้นตอนการทำงาน ดังนี้

- 1) เมื่อพบคำสั่ง if และตรวจสอบเงื่อนไขพบว่า เป็นเท็จแล้ว จะเก็บคำสั่ง if ไว้ใน statementNotDone
  - 2) ชุดคำสั่งที่อยู่ภายใต้คำสั่ง if ไม่ถูกทำงาน
  - 3) เมื่อหมดชุดคำสั่ง โปรแกรมจะกำจัดค่าที่เก็บไว้ที่ statementNotDone ซึ่งเป็นค่าล่าสุดออกไป
- จากขั้นตอนข้างต้น สามารถอธิบายได้ดังรูปที่ 4-4 ซึ่งเป็นการแสดงการทำงานของโปรแกรม ที่เขียนโดยคำสั่งสคริปต์ ดังนี้ (กำหนดให้  $x = 1$ )

```
if (x>2) then
begin
    ??? (ชุดคำสั่ง)
end;
```



รูปที่ 4-4 แสดงการทำงานของคำสั่ง if เมื่อเงื่อนไขเป็นเท็จ

#### ● IF...ELSE statement

การอิมพลิเมนต์เพื่อจัดการคำสั่ง IF...ELSE ในภาษาสคริปต์ จะคล้ายกับการอิมพลิเมนต์คำสั่ง if เพียงแต่ว่า การจัดการคำสั่ง IF...ELSE นั้น จะต้องมีชุดคำสั่งชุดหนึ่งที่ไม่ได้ถูกทำงาน ภายใต้คำสั่งของ if หรือ else นอกจากนี้ ตัวแปรที่มีบทบาทต่อการจัดการอีกตัวคือ ตัวแปร ifNotDone ชนิดบูลีน (boolean) ทำให้โปรแกรมรู้ว่าควรจะทำชุดคำสั่งใด ระหว่างชุดคำสั่งภายใต้ if หรือ else ซึ่งความเป็นไปได้ของงานแบ่งเป็น 2 กรณี คือ กรณีที่เงื่อนไขของ if ถูกต้อง และในกรณีที่เงื่อนไขของ if ไม่ถูกต้อง ในกรณีที่คำสั่ง if ทำงานถูกต้องมีขั้นตอนการทำงานเป็นดังนี้

- 1) เมื่อพบคำสั่ง if และเงื่อนไขเป็นจริง จะเก็บ if ไว้ที่ statementDone รวมทั้งกำหนดให้ ifNotDone เป็นเท็จ
- 2) ทำชุดคำสั่งของ if
- 3) เมื่อหมดชุดคำสั่งของ if ให้กำจัดค่าที่เก็บใน statementDone ที่อยู่ล่าสุดออก
- 4) เมื่อพบคำสั่ง else ทำการตรวจสอบ ifNotDone เมื่อพบว่าเป็นเท็จ จะเก็บ else ไว้ที่ statementNotDone
- 5) ชุดคำสั่งของ else จะไม่ถูกทำงาน
- 6) เมื่อหมดชุดคำสั่งของ else ต้องกำจัดค่าที่เก็บใน statementNotDone ที่อยู่ล่าสุดออก

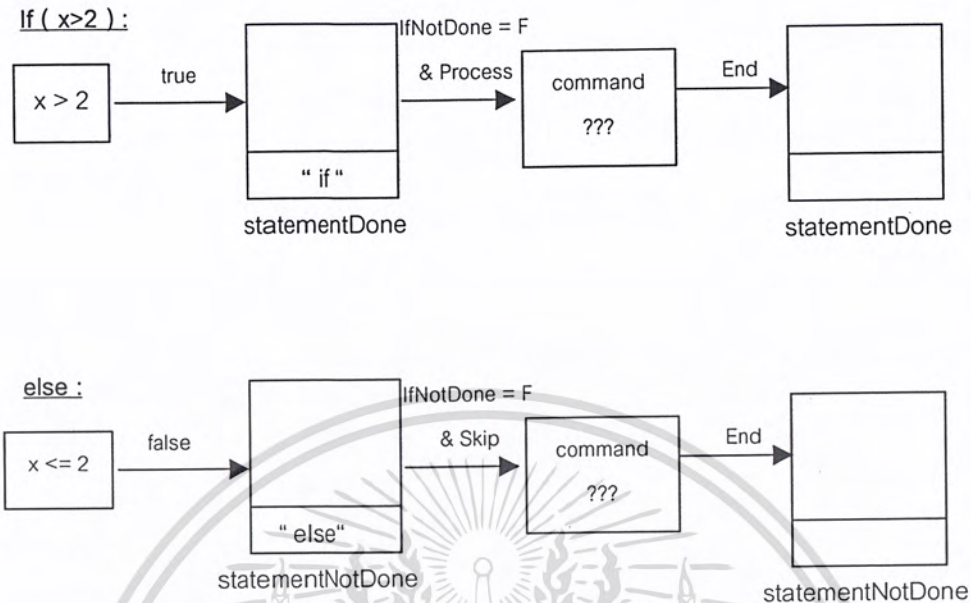
จากขั้นตอนข้างต้น สามารถอธิบายได้ดังรูปที่ 4-5 ซึ่งเป็นการแสดงการทำงานของโปรแกรม ที่เขียนโดยคำสั่งสคริปต์ ดังนี้ (กำหนดให้  $x = 4$ )

```

if (x>2) then
    begin
        ??? (ชุดคำสั่ง)
    end;
else
    begin
        xxx
    end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-5 แสดงการทำงานของคำสั่ง if-else

ในอีกกรณีหนึ่ง คือ เงื่อนไขของคำสั่ง if เป็นเท็จ กรณีนี้โปรแกรมจะไม่ทำชุดคำสั่งของ if แต่จะทำชุดคำสั่งของ else แทน ซึ่งขั้นตอนจะคล้ายกับกรณีแรก แตกต่างเพียง ifNotDone จะมีค่าเป็นจริง ทำให้โปรแกรมไม่ทำชุดคำสั่งของ if และข้อควรระวังตอนสุดท้ายคือ เมื่อชุดคำสั่งของ else หมดแล้ว ต้องเปลี่ยนค่า ifNotDone ให้เป็นเท็จ เพราะในการทำงานครั้งใหม่ผ่าน IF...ELSE จะต้องเริ่มต้นที่คำสั่ง if แล้วจึงตรวจสอบเงื่อนไข เพื่อกำหนดค่า ifNotDone ต่อไป

#### 4.2.3 ความสามารถในการวนรอบ

คำสั่งที่จัดการเกี่ยวกับการวนรอบสำหรับภาษาสคริปต์ที่ผู้จัดทำโครงการได้กำหนดขึ้นนั้น มี 2 คำสั่ง ดังนี้

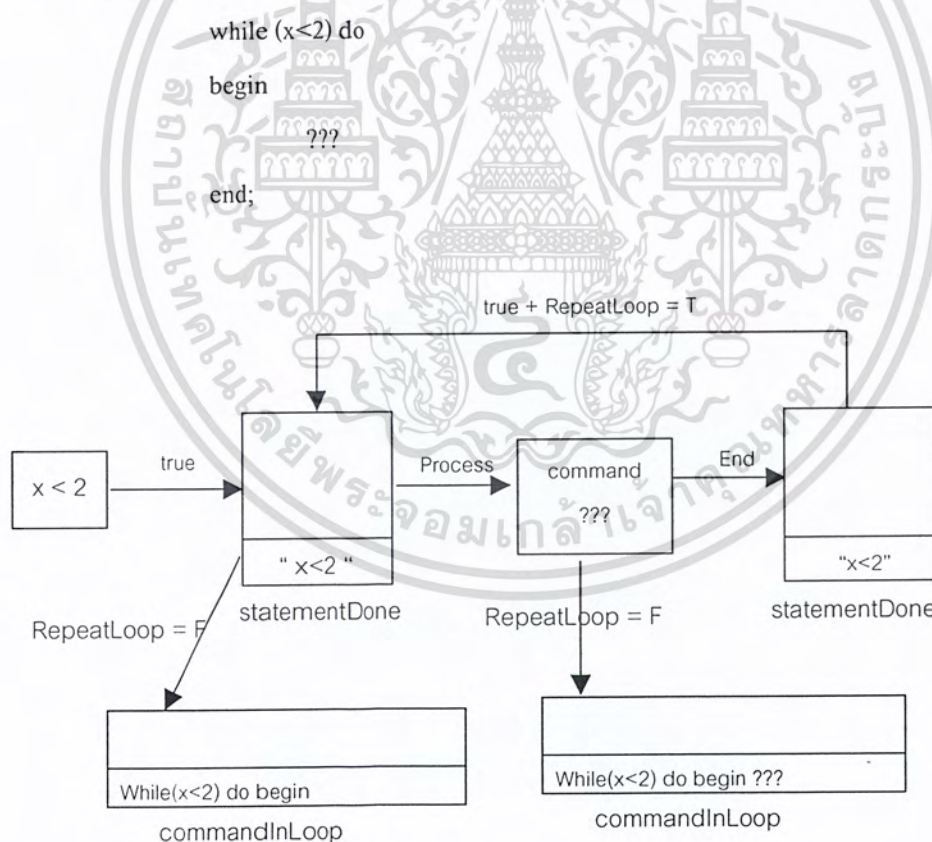
- While statement

การอิมพลิเมนต์การทำงานของคำสั่ง while นั้นต้องอาศัยตัวแปรสำคัญอีกตัวหนึ่งเพื่อเพิ่มความสามารถให้ทำการวนรอบได้ นั่นคือ ตัวแปร commandInLoop ชนิดเวกเตอร์ ทำหน้าที่ในการเก็บชุดคำสั่งทั้งหมด ของคำสั่งที่มีโอกาสที่จะวนรอบได้ นอกจากนี้ การเก็บคำสั่งใน statementDone และ statementNotDone ก็แตกต่างกันไปจากที่กำหนดในคำสั่ง if นั่นคือ แทนที่จะเก็บ while ลงไป แต่จะเก็บเป็นเงื่อนไขของคำสั่ง while เพื่อสามารถตรวจสอบเงื่อนไข และเป็นประโยชน์ต่อการวนรอบต่อไป การ

ทำงานของคำสั่ง while แบ่งออกได้เป็น 2 แบบ คือ กรณีแรก เงื่อนไขของคำสั่ง while เป็นจริง และกรณีที่สอง คือ เงื่อนไขของคำสั่ง while เป็นเท็จ ในกรณีที่สอง มีหลักการทำงานคล้ายกับคำสั่ง if ข้างต้น จึงไม่ขอกล่าวถึง แต่สำหรับกรณีแรกนั้น มีขั้นตอนการทำงานดังนี้

- 1) เมื่อพบคำสั่ง while และเงื่อนไขของคำสั่งเป็นจริง เก็บเงื่อนไขของคำสั่งไว้ใน statementDone
- 2) ทำชุดคำสั่ง ภายในคำสั่ง while และเมื่อทำงานแต่ละคำสั่งเสร็จแล้ว ให้นำคำสั่งนั้นไปต่อกับคำสั่งเดิมที่อยู่ใน commandInLoop ที่อยู่ตำแหน่งล่าสุด ถ้ายังไม่มีคำสั่งใด ๆ เลย ก็ทำการสร้างขึ้นมาเอง
- 3) เมื่อชุดคำสั่งหมด ต้องไปตรวจดูเงื่อนไขบน statementDone ว่าเงื่อนไขยังเป็นจริงอยู่หรือไม่ ถ้าเป็นจริง ก็ทำขั้นตอนเดิมต่อไปเรื่อย ๆ เพียงแต่ไม่ต้องเพิ่มคำสั่งเดิมเข้าไปเก็บอีก โดยมีตัวแปร RepeatLoop เป็นตัวกำหนดให้ แต่ถ้าหากเงื่อนไขเป็นเท็จ ก็สามารถออกจากการทำงานวนรอบได้ ตามการทำงานที่อธิบายไปแล้ว

จากขั้นตอนข้างต้น สามารถอธิบายได้ดังรูปที่ 4-6 ซึ่งเป็นการแสดงการทำงานของโปรแกรม ที่เขียนโดยคำสั่งสคริปต์ ดังนี้ (กำหนดให้  $x = 1$ )



รูปที่ 4-6 แสดงการทำงานของคำสั่ง while

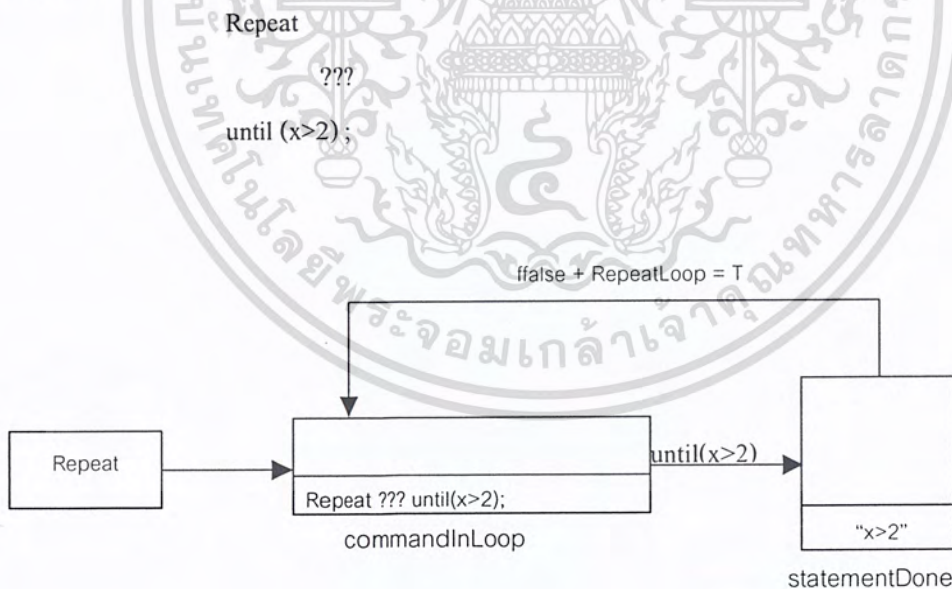
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Repeat...Until statement

ลักษณะการอิมพลิเมนต์ของคำสั่ง Repeat...Until จะคล้ายกับการอิมพลิเมนต์ของคำสั่ง while ต่างที่การตรวจสอบเงื่อนไขของคำสั่ง คือการทำงานแบบ Repeat...Until จะทำหุคคำสั่งก่อน แล้วจึงทำการตรวจสอบเงื่อนไขทีหลัง ดังนั้น หุคคำสั่งจะถูกทำอย่างน้อย 1 ครั้งเสมอ และข้อแตกต่างอีกอย่างหนึ่งคือ ลักษณะการออกจากการทำงานวนรอบเนื่องจากการตรวจสอบเงื่อนไข การทำงานแบบ Repeat...Until นี้ เมื่อเงื่อนไขเป็นจริง จะออกจากการทำวนรอบ แต่ถ้าเงื่อนไขเป็นเท็จจะวนรอบ ในกรณีทีเงื่อนไขเป็นเท็จ จะมีเงื่อนไขการทำงานดังนี้

- 1) เมื่อพบคำสั่ง Repeat จะทำงานหุคคำสั่งได้เลย
- 2) การทำงานแต่ละอย่างของหุคคำสั่ง จะถูกนำไปต่อกับคำสั่งเดิม ใน commandInLoop หรือ หากยังไม่มีคำสั่งใด ๆ เลย ต้องทำการสร้างขึ้นเอง เหมือนกับการจัดการกับคำสั่ง while
- 3) เมื่อพบคำสั่ง Until จะทำการตรวจสอบเงื่อนไข และเก็บเงื่อนไขไว้ใน statementDone เพื่อใช้ในการตรวจสอบต่อไป ถ้าเป็นเท็จ จะทำหุคคำสั่งที่เก็บไว้ใน statementDone อีกครั้ง แต่ไม่ต้องนำคำสั่งไปต่อกับคำสั่งเดิมที่เก็บไว้ใน statementDone อีก โดยใช้ตัวแปร Repeat Loop กำหนดการทำงานนี้ แต่ถ้าเงื่อนไขเป็นจริงจะออกจากการวนรอบ

จากขั้นตอนข้างต้น สามารถอธิบายได้ด้วยรูปที่ 4-7 ซึ่งเป็นการแสดงการทำงานของโปรแกรม ที่เขียนโดยคำสั่งสคริปต์ ดังนี้ (กำหนดให้  $x = 1$ )



รูปที่ 4-7 แสดงการทำงานของคำสั่ง repeat-until

### 4.3 การทำงานของเอเจนต์

ส่วนประกอบสำคัญของการอิมพลิเมนต์การเขียนโปรแกรม มีดังต่อไปนี้

#### 4.3.1 ความสามารถทำงานแบบเชรดได้

เป็นความสามารถที่สนับสนุนเอเจนต์ให้ทำงานได้โดยอัตโนมัติ และเป็นอิสระจากการทำงานอื่น การกำหนดคุณสมบัติของเอเจนต์ให้ทำงานแบบเชรด เป็นการสนับสนุนการทำงานโดยรวมของบราวเซอร์ที่สามารถโปรแกรมได้ โดยเฉพาะการทำงานของบราวเซอร์ที่เกี่ยวข้องกับเวลา เช่นการตั้งเวลา เพื่อให้ทำงานอย่างหนึ่ง ปล่อยให้บราวเซอร์ทำงานตามเวลาที่กำหนด เป็นต้น

การอิมพลิเมนต์โปรแกรม เพื่อให้เอเจนต์มีการทำงานแบบเชรดได้ ทำได้โดยการใช้คลาสเชรด (Thread) ของภาษาจาวา รวมทั้งการอิมพลิเมนต์ส (Implements) อินเทอร์เฟซ รันเนเบิล (Interface Runnable) ซึ่งมีให้แล้วในภาษาจาวา

#### 4.3.2 ความสามารถในการจัดการด้านเวลา

เอเจนต์ต้องมีความสามารถในการจัดการด้านเวลา เพื่อประโยชน์ในการตั้งเวลาให้บราวเซอร์ที่สามารถโปรแกรมได้ ทำงานได้อย่างเป็นอิสระ และอัตโนมัติ การอิมพลิเมนต์โปรแกรมที่สนับสนุนการตั้งเวลา หรือการทำงานที่เกี่ยวข้องกับเวลาได้ ต้องอาศัยคลาสคาเลนดาร์ (Calendar) ของภาษาจาวา เพื่อใช้ในการบอกวันที่ และเวลา และสนับสนุนการตั้งเวลาโดยตรง

วิธีการอิมพลิเมนต์สำหรับจัดการการตั้งเวลา มีลำดับขั้นตอนดังนี้

- 1) ผู้ใช้ต้องระบุเวลาที่ต้องการให้บราวเซอร์ทำงานต่าง ๆ ให้แก่เอเจนต์
- 2) เอเจนต์ทำงาน
- 3) เอเจนต์ทำการค้นหาเวลาปัจจุบัน โดยใช้เมธอด getInstance() ของคลาส Calendar เพื่อหาเวลาในหน่วยชั่วโมง, นาที และ วินาที
- 4) คำนวณหาเวลาที่เหลือเพื่อทำงานที่กำหนดไว้ โดยนำเวลาที่ผู้ใช้ตั้งไว้ ลบด้วยเวลาในปัจจุบัน แล้วคำนวณความแตกต่างที่ได้ให้อยู่ในหน่วยของมิลลิวินาที
- 5) ใช้เมธอด sleep(different time) ของคลาส Thread ให้หยุดทำงานชั่วคราว จนถึงเวลาที่กำหนด (ความแตกต่างในหน่วยมิลลิวินาที จากข้อที่แล้ว) Thread จะทำการปลุกให้เอเจนต์ทำงานได้

#### 4.3.3 ความสามารถในการจัดการการทำงาน

การทำงานหลักของบราวเซอร์ที่สามารถโปรแกรมได้นั้น แบ่งเป็น 2 ส่วน คือ

- 1) การทำงานเกี่ยวกับบราวเซอร์

บราวเซอร์ที่นำมาแสดงต่อผู้ใช้ เป็นบราวเซอร์ซึ่งเขียนขึ้นโดยภาษาจาวา และใช้ Swing Set Component สนับสนุนการทำงานผ่านเครือข่ายด้วยโพรโทคอลเอชทีทีพี บราวเซอร์ที่สร้างขึ้นถูกเขียนเป็นโปรแกรมจาวาแอปพลิเคชัน ผู้ใช้สามารถกำหนดเวลาในการเปิดและปิดบราวเซอร์ได้ ผ่านเมธอด sleep() ให้หยุดการทำงานระยะเวลาหนึ่ง และปลุกให้มาทำงานเมื่อถึงเวลาที่กำหนด ซึ่งงานในที่นี้หมายถึงการเปิด หรือปิดบราวเซอร์นั่นเอง

เอเจนต์ทำงานเกี่ยวข้องกับบราวเซอร์ได้ 2 แบบดังนี้

- เปิดบราวเซอร์

ผู้จัดทำโครงการได้กำหนด ให้มีตัวแปรในการนับจำนวนบราวเซอร์ในการเปิดบราวเซอร์แต่ละครั้ง รวมทั้งได้กำหนดให้ผู้ใช้สามารถเปิดบราวเซอร์มาดูได้ไม่เกิน 10 บราวเซอร์ ซึ่งเอเจนต์จะเก็บบราวเซอร์ต่าง ๆ ลงในอาร์เรย์ เอเจนต์เปิดบราวเซอร์ได้โดยยูอาร์แอลที่ผู้ใช้กำหนดให้ หรือในทางตรงข้าม หากผู้ใช้ไม่ได้ระบุยูอาร์แอลที่ต้องการติดต่อ เมื่อเปิดบราวเซอร์ขึ้น จะปรากฏยูอาร์แอลซึ่งเป็นมาตรฐานที่ผู้จัดทำโครงการได้กำหนดขึ้น

- ปิดบราวเซอร์

การปิดบราวเซอร์แต่ละครั้ง จะทำการลดค่าของตัวแปรที่นับจำนวนบราวเซอร์ เพื่อให้สามารถตรวจสอบว่าปัจจุบันมีบราวเซอร์เปิดอยู่ทั้งหมดกี่ตัว

## 2) การทำงานเกี่ยวกับซีจีไอ

การทำงานเกี่ยวกับซีจีไอที่ได้อธิบายถึงนี้ เป็นความสามารถที่ผู้ใช้สามารถป้อนลงไปคำสั่งสคริปต์ได้ หลักการโดยรวมคือ การมองซีจีไอให้เสมือนเป็นฟังก์ชันหรือโพรซีเจอร์ตัวหนึ่ง ซึ่งสามารถส่งค่าพารามิเตอร์ให้แก่ซีจีไอได้ รวมทั้งซีจีไอสามารถส่งค่ากลับมาให้แก่ผู้ใช้ได้ เสมือนเป็นฟังก์ชันในโปรแกรมทั่วไป ฟังก์ชันหนึ่ง ซึ่งการทำงานของซีจีไอแบ่งเป็น 2 ส่วนดังนี้

- การทำงานแบบเก็ท

ผู้จัดทำโครงการได้กำหนดงานที่สามารถทำงานเกี่ยวกับซีจีไอแบบเก็ทขึ้นมาหนึ่งงาน คือ การค้นหาเว็บเพจตามคีย์เวิร์ดที่ผู้ใช้ต้องการ โดยผู้จัดทำโครงการจะเรียกบริการซีจีไอจากเซิร์ฟเวอร์ตัวหนึ่งซึ่งผู้จัดทำโครงการเป็นผู้กำหนดขึ้นเอง การส่งข้อมูลเพื่อทำงานในลักษณะนี้เป็นการทำงานแบบเก็ท โดยจะส่งคีย์เวิร์ดพ่วงไปกับยูอาร์แอลด้วย

ตัวอย่างเช่น ผู้ใช้ส่งคำสั่ง search(computer) และผู้จัดทำโครงการกำหนดให้ใช้โปรแกรมการค้นหา (Search Engine) ของ <http://www.yahoo.com> ซึ่งซีจีไอของโปรแกรมนี้คือ <http://search.yahoo.com/bin/query> ดังนั้นเอเจนต์จะทำการเชื่อมต่อเครือข่ายผ่านยูอาร์แอล <http://search.yahoo.com/bin/query?p=computer> ซึ่งแสดงการทำงานได้ดังรูป

- การทำงานแบบโพสต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้จัดทำโครงการได้กำหนดงานที่สามารถทำงานเกี่ยวกับซีจีไอแบบโพสต์ชันมาหนึ่งงาน คือ การส่งข้อความผ่านบริการทางเพจเจอร์ โดยผู้จัดทำโครงการได้กำหนดเซิร์ฟเวอร์เพื่อเรียกใช้บริการซีจีไอเอาไว้แล้ว และการส่งข้อมูลของซีจีไอแบบนี้ต้องส่งผ่านอินพุทเอาต์พุตมาตรฐาน โดยข้อมูลที่จะส่งต้องผ่านการเข้ารหัส (Encode) ก่อน

ตัวอย่างเช่น ผู้ใช้ส่งคำสั่ง `page(162,254587,7777,hello)` หมายถึงผู้ใช้ส่งข้อความว่า "hello" ไปที่หมายเลข 162-254587 และมีรหัสผ่าน 7777 ข้อมูลเหล่านี้ต้องผ่านการเข้ารหัสก่อน แต่สิ่งแรกที่ต้องเชื่อมต่อไปที่เซิร์ฟเวอร์ผู้ให้บริการซีจีไอเสียก่อน ซึ่งผู้จัดทำโครงการได้กำหนดไว้แล้ว

#### 4.4 การส่งและรับข่าวของบราวเซอร์ที่สามารถโปรแกรมได้ผ่าน NNTP

ส่วนประกอบสำคัญของการอิมพลีเมนต์การเขียนโปรแกรม มีดังต่อไปนี้

##### 4.4.1 จัดการการสื่อสารกับเอเจนต์ตัวอื่น

บราวเซอร์ที่สามารถโปรแกรมได้กำหนดความสามารถในการรับส่งข่าวสารผ่านโทรโตคอลเอ็นเอ็นทีพีได้ และเมื่อได้รับข่าวสารแล้วผู้รับสามารถส่งข่าวไปให้แก่คนอื่นได้อีก โดยผ่านการจัดการของเอเจนต์ ทั้งฝ่ายรับและส่งข่าว การอิมพลีเมนต์โปรแกรมที่สนับสนุนการสื่อสารกับเอเจนต์ตัวอื่น โดยอาศัยหลักการของรีโมตเมธอดอินโวกชัน เพื่อสนับสนุนการส่งข้อมูลหรือเรียกใช้บริการจากระยะไกล ดังนี้

- 1) โปรแกรมที่ฝังเอเจนต์ (ผู้ใช้) ต้องสามารถทำการอ้างอิงวัตถุระยะไกลได้ (ในที่นี้หมายถึงเซิร์ฟเวอร์ที่ให้บริการแก่เอเจนต์ทั้งหลาย) โดยใช้เมธอด `lookup(URL)` ของคลาส `Naming` ซึ่งยูอาร์แอลจะเป็นสิ่งที่ระบุถึงโฮสต์ และต้องขึ้นต้นด้วย `"rmi://"`
- 2) เอเจนต์ต้องทำการลงทะเบียนแก่เซิร์ฟเวอร์ โดยเรียกใช้เมธอด `register()` เพื่อส่งตัวเองและชื่อไปให้แก่เซิร์ฟเวอร์ ซึ่งเมธอดนี้เป็นเมธอดของฝั่งเซิร์ฟเวอร์ เพื่อประโยชน์ในการรับข่าวสารจากเซิร์ฟเวอร์
- 3) เอเจนต์ต้องมีไฟล์บอกรายละเอียดของตัวเอง เรียกว่ายูสเซอร์โพรไฟล์ซึ่ง ประกอบด้วยชื่อของเอเจนต์, ประเภทข่าวที่เอเจนต์สนใจ และ ชื่อของเอเจนต์ตัวอื่นที่ต้องการติดต่อด้วย หากเอเจนต์ต้องการติดต่อกับเอเจนต์ตัวอื่น จำเป็นต้องมียูสเซอร์โพรไฟล์ด้วย เพื่อระบุปลายทางของการส่งข่าว และต้นทางคือผู้ส่ง
- 4) ที่ฝั่งเอเจนต์ต้องมีเมธอด `receive()` เพื่อสนับสนุนการรับข่าวสารที่มาจากเอเจนต์ตัวอื่น โดยเซิร์ฟเวอร์เป็นตัวจัดการส่งข่าวสารนั้นมาให้ กำหนดให้เอเจนต์จัดการข่าวสารที่ได้เก็บเป็นไฟล์ และมีการเตือนผู้ใช้ว่า ได้รับข่าวสารจากระยะไกล
- 5) เอเจนต์ต้องมีเมธอด `getStatus()` เพื่อส่งประเภทข่าวที่สนใจให้แก่เซิร์ฟเวอร์ได้ ในกรณีที่มิเอเจนต์ตัวอื่นส่งข่าวสารมาให้ ซึ่งเซิร์ฟเวอร์ต้องจัดการและพิจารณาการส่งข่าวสารนั้นจากประเภทข่าวที่สนใจของเอเจนต์ที่จะได้รับข่าวนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเอเจนต์ได้รับข่าวสารของคนและส่งข่าวนั้นให้เอเจนต์ตัวอื่น เอเจนต์ต้องเรียกเมธอด `sendMessage()` ที่ฝั่งเซิร์ฟเวอร์ เพราะเซิร์ฟเวอร์ทำหน้าที่จัดการการรับส่งข่าวสารจากเอเจนต์ต่าง ๆ ทั้งยังเป็นผู้พิจารณาว่าส่งข่าวให้เอเจนต์ปลายทางได้หรือไม่ โดยการเรียกเมธอด `getStatus()` จากฝั่งเอเจนต์ และตรวจสอบว่าขณะนั้น เอเจนต์ปลายทางทำงานอยู่หรือไม่

#### 4.5 การทำงานของบราวเซอร์ที่สามารถโปรแกรมได้ผ่านโปรแกรมเทลเน็ต (Telnet)

ส่วนประกอบสำคัญของการอิมพลิเมนต์การเขียนโปรแกรม มีดังต่อไปนี้

##### 4.5.1 การจัดการการทำงานผ่านโปรแกรมเทลเน็ต

การใช้งานบราวเซอร์ที่สามารถโปรแกรมได้ นอกจากจะถูกทำงานจากการสั่งงานระยะไกลแล้ว ยังสามารถถูกสั่งงานจากผู้ใช้ในระยะไกลได้อีกด้วย โดยการจัดการในการทำงานก็แตกต่างออกไป ดังต่อไปนี้

- 1) เมื่อเปิดบราวเซอร์ที่สามารถโปรแกรมได้ขึ้น โปรแกรมจะทำการเปิดเซิร์ฟเวอร์สำหรับให้บริการการสั่งงานจากระยะไกลขึ้นทันที
- 2) ผู้ใช้ที่ต้องการสั่งงานบราวเซอร์ผ่านโปรแกรมเทลเน็ต ต้องเชื่อมต่อไปยังโฮสต์และพอร์ตที่ถูกตั้ง ซึ่งได้กำหนดให้ติดต่อผ่านพอร์ต 6666
- 3) เมื่อเซิร์ฟเวอร์ได้รับการติดต่อจากผู้ใช้ในระยะไกล ต้องสร้างวัตถุที่ทำหน้าที่จัดการการเชื่อมต่อระหว่างผู้ใช้งานกับเซิร์ฟเวอร์หรือบราวเซอร์ที่สามารถโปรแกรมได้ ซึ่งเรียกวัตถุนั้นว่า เทลเน็ตแฮนเดเลอร์ (Telnet Handler)

## บทที่ 5

### การทำงานของบราวเซอร์

#### 5.1 การใช้งานบราวเซอร์ที่สามารถโปรแกรมด้วยคำสั่งสคริปต์

เมื่อเปิดโปรแกรมบราวเซอร์ที่สามารถโปรแกรมได้ขึ้นมาจะพบกับหน้าต่างหลักของโปรแกรม ดังรูปที่ 5-1



รูปที่ 5-1 รูปแสดงหน้าต่างหลักของโปรแกรม

พื้นที่ที่เท็กซ์แเอเรียที่อยู่ในหน้าต่างหลักของโปรแกรมนี้นั้นเป็นพื้นที่ ที่ใช้สำหรับเขียนสคริปต์ เมื่อผู้เขียนเขียนสคริปต์เสร็จแล้วผู้เขียนสามารถสั่งให้โปรแกรมทำงานตามสคริปต์ที่เขียนได้ คำสั่งสคริปต์ที่สามารถสั่งงานบราวเซอร์ได้มีดังนี้

##### 1. คำสั่งเปิดบราวเซอร์

open()	เป็นคำสั่งสคริปต์ที่ใช้เปิดบราวเซอร์โดยผู้เขียน ไม่ระบุยูอาร์แอล บราวเซอร์ที่เปิดขึ้นมาจะแสดงเว็บเพจของยูอาร์แอลที่กำหนดเอาไว้ในโปรแกรม ซึ่งยูอาร์แอลที่กำหนดเอาไว้ในโปรแกรม คือ <a href="http://www.download.com">http://www.download.com</a>
open(URL)	เป็นคำสั่งสคริปต์ที่ใช้เปิดบราวเซอร์โดยระบุยูอาร์แอลของไซต์ที่ต้องการ บราวเซอร์ที่เปิดขึ้นมาจะแสดงเว็บเพจของ URL ที่ผู้เขียนสคริปต์ระบุเอาไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2. คำสั่งปิดบราวเซอร์

Close	เป็นคำสั่งสคริปต์ที่ใช้ปิดบราวเซอร์ ในกรณีที่มีบราวเซอร์เปิดเอาไว้อยู่แล้วมากกว่า 1 หน้าต่าง คำสั่งนี้จะปิดบราวเซอร์ตัวล่าสุดเพียงหน้าต่างเดียว
-------	---

## 3. คำสั่งสคริปต์ที่สั่งให้บราวเซอร์ไปค้นหาคำที่ผู้เขียนสคริปต์ต้องการ

Search(keyword)	เป็นคำสั่งสคริปต์ที่ใช้สั่งงานบราวเซอร์ โดยคำสั่งนี้จะสั่งให้บราวเซอร์ไปค้นหา keyword ที่ผู้เขียนสคริปต์ต้องการ โดยจะทำการบันทึกไฟล์ผลลัพธ์ที่ได้จากการค้นหาเก็บไว้ในไฟล์ D:\search_result.html
-----------------	---

นอกจากนี้ยังมีคำสั่งพิเศษที่เกิดจากการมอดูอาร์แอลให้เป็นฟังก์ชัน จากแนวคิดนี้เองทำให้สามารถสั่งงานบราวเซอร์ได้โดยเขียนยูอาร์แอลที่มีการออกแบบให้เป็นฟังก์ชัน แล้วส่งข้อมูลต่าง ๆ ที่ฟังก์ชันนั้นต้องการไปยังยูอาร์แอลที่มีฟังก์ชันการทำงานนั้นอยู่ แล้วรับผลลัพธ์จากการทำงานของฟังก์ชันกลับมา

## 4. คำสั่งสคริปต์คำนวณอัตราแลกเปลี่ยนเงินจากเงินบาทเป็นเงินสกุลต่าง ๆ

161.246.4.3/~s0010892/b2fc(DC,BA)	เป็นคำสั่งสคริปต์ที่ใช้สั่งงานบราวเซอร์ให้เรียกใช้บริการฟังก์ชันคำนวณอัตราแลกเปลี่ยนเงินตรา จากเงินบาทเป็นเงินสกุลต่าง ๆ โดย - DC (Desired Currency) คือ สกุลเงินที่ต้องการ สามารถดูรายละเอียดได้จากตารางที่ 5-1 - BA (Baht Amount) คือ จำนวนเงิน (หน่วยบาท) ที่ต้องการเปลี่ยน
-----------------------------------	--

## 5. คำสั่งสคริปต์คำนวณอัตราแลกเปลี่ยนเงินจากเงินสกุลต่าง ๆ เป็นเงินบาท

161.246.4.3/~s0010892/fc2b(DC,FA)	เป็นคำสั่งสคริปต์ที่ใช้สั่งงานบราวเซอร์ให้เรียกใช้บริการฟังก์ชันคำนวณอัตราแลกเปลี่ยนเงินตรา จากเงินสกุลต่าง ๆ เป็นเงินบาท โดย - DC (Desired Currency) คือ สกุลเงินที่ต้องการ สามารถดูรายละเอียดได้จากตารางที่ 5-1
-----------------------------------	--

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	- <i>FA</i> (Foreign Currency Amount) คือ จำนวนเงิน (ตามสกุลที่ระบุโดย <i>DC</i> ) ที่ต้องการเปลี่ยน
--	--

ตารางที่ 5-1 ตารางแสดงตัวย่อของ *Currency* และความหมาย

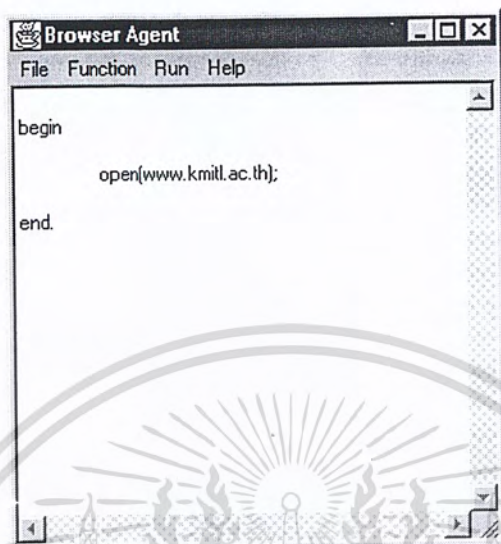
Currency	COUNTRY
USD	US Dollar
EUR	EURO
GBP	GRATE BRITAIN
DEM	GERMANY
JPY	JAPAN
FRF	FRANCE
AUD	AUSTRALIA
CNY	CHINA
HKD	HONG KONG
CAD	CANADA

6. คำสั่งสกริปต์ในการส่งเพจ

161.246.4.3/~s0010892/page( <i>CP,NO,PW,MSG</i> )	<p>เป็นคำสั่งสกริปต์ที่ใช้ส่งข้อความไปยังเพจเจอร์</p> <ul style="list-style-type: none"> <li>- <i>CP</i> (Company) ผู้ให้บริการเพจเจอร์ต่าง ๆ เช่น 142, 152, 162, 1144, 1188</li> <li>- <i>NO</i> (Number) หมายเลขเครื่องเพจเจอร์ที่ต้องการส่งข้อความไปให้</li> <li>- <i>PW</i> (Password) พาสเวิร์ดสำหรับเพจเจอร์ของ 162</li> <li>- <i>MSG</i> (Message) ข้อความที่ต้องการส่ง</li> </ul>
---	---

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพตัวอย่างของสคริปต์ที่สามารถสั่งให้โปรแกรมทำงานได้  
ตัวอย่างการเขียนสคริปต์เพื่อเปิดบราวเซอร์



รูปที่ 5-2 แสดงการเขียนสคริปต์เพื่อเปิดบราวเซอร์

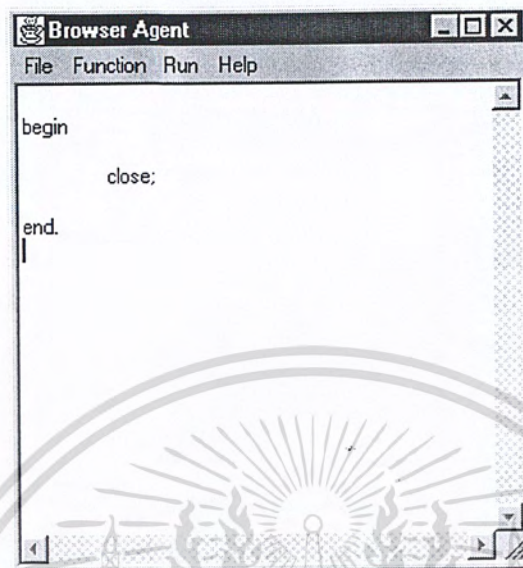
จากภาพจะเห็นได้ว่าเป็นการเปิดบราวเซอร์ไปยังเว็บเพจของ www.kmitl.ac.th



รูปที่ 5-3 แสดงเว็บเพจของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ตัวอย่างการเขียนสคริปต์เพื่อปิดบราวเซอร์

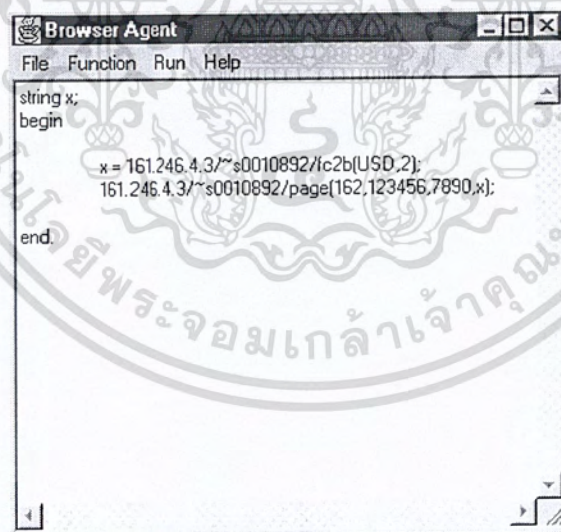


```

Browser Agent
File Function Run Help
begin
    close;
end.
  
```

รูปที่ 5-4 แสดงการเขียนสคริปต์เพื่อปิดบราวเซอร์

### ตัวอย่างการเขียนสคริปต์เพื่อคำนวณอัตราแลกเปลี่ยนเงินตราและส่งเพจ



```

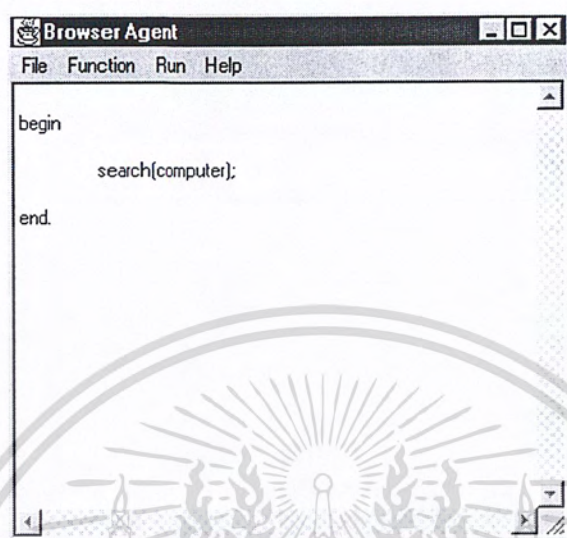
Browser Agent
File Function Run Help
string x;
begin
    x = 161.246.4.3/~s0010892/ic2b(USD,2);
    161.246.4.3/~s0010892/page(162,123456,7890,x);
end.
  
```

รูปที่ 5-5 แสดงการเขียนสคริปต์เพื่อคำนวณอัตราแลกเปลี่ยนเงินตราและส่งเพจ

จากภาพจะเห็นได้ว่าเป็นการเรียกใช้ฟังก์ชันในการคำนวณอัตราแลกเปลี่ยนเงินจากเงินสกุล USD เป็นเงินบาท และส่งผลลัพธ์จากการคำนวณไปยังเพจเจอร์โดยเรียกใช้ฟังก์ชันในการส่งเพจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการเขียนสคริปต์เพื่อค้นหาข้อมูลที่ต้องการ

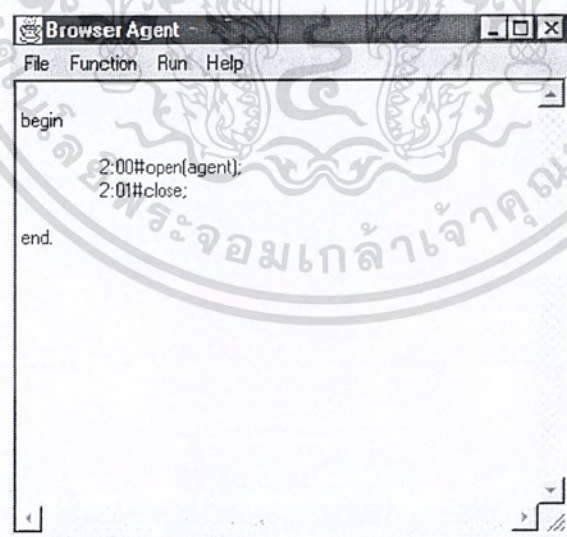


```

Browser Agent
File Function Run Help
begin
    search(computer);
end.
  
```

รูปที่ 5-6 แสดงการเขียนสคริปต์เพื่อค้นหาข้อมูลที่ต้องการ

จากภาพจะเห็นว่าเป็นการเรียกใช้ฟังก์ชัน ในการค้นหาข้อมูล คำที่ต้องการค้นหา คือ Computer  
ตัวอย่างการเขียนสคริปต์สั่งงานตามเวลาที่กำหนด



```

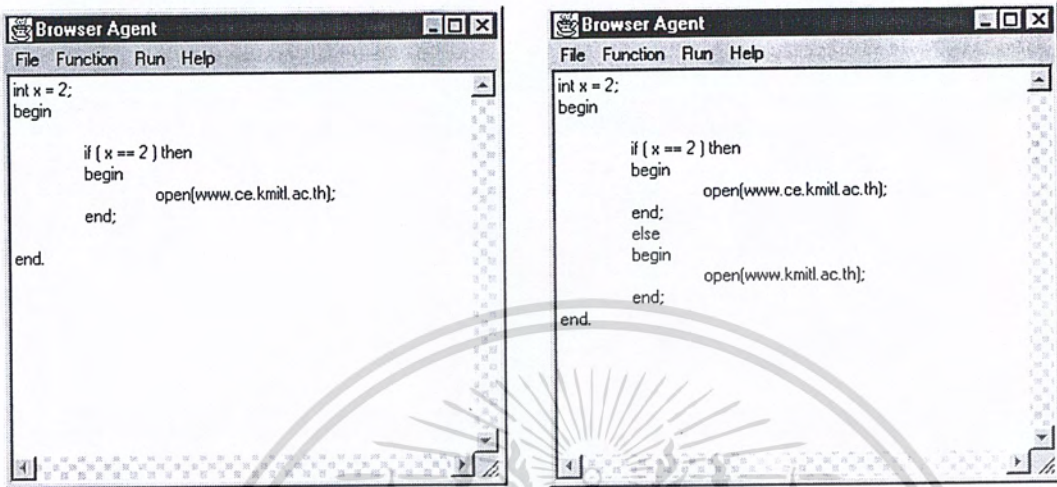
Browser Agent
File Function Run Help
begin
    2:00#open(agent);
    2:01#close;
end.
  
```

รูปที่ 5-7 แสดงการเขียนสคริปต์สั่งให้เบราว์เซอร์เปิดและปิดตามเวลาที่กำหนด

จากภาพเป็นการเขียนสคริปต์สั่งให้เบราว์เซอร์เปิดและปิดตามเวลาที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการเขียนสคริปต์สั่งงานแบบมีเงื่อนไข



```

int x = 2;
begin

    if { x == 2 } then
    begin
        open(www.ce.kmitl.ac.th);
    end;
end.

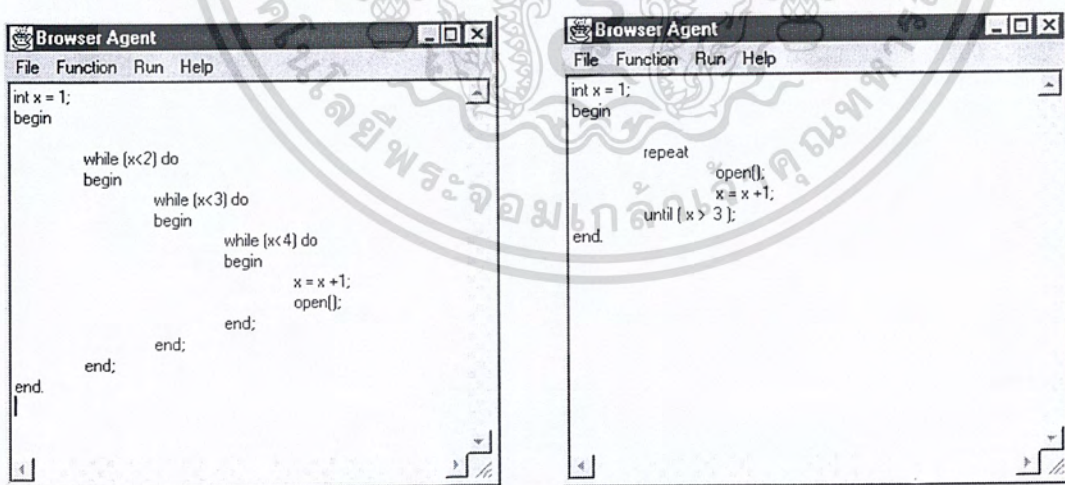
int x = 2;
begin

    if { x == 2 } then
    begin
        open(www.ce.kmitl.ac.th);
    end;
    else
    begin
        open(www.kmitl.ac.th);
    end;
end.

```

รูปที่ 5-8 แสดงเงื่อนไข if และ if-else

ตัวอย่างการเขียนสคริปต์สั่งงานแบบวนลูปแบบต่าง ๆ



```

int x = 1;
begin

    while {x<2} do
    begin
        while {x<3} do
        begin
            while {x<4} do
            begin
                x = x + 1;
                open();
            end;
        end;
    end;
end.

int x = 1;
begin

    repeat
        open();
        x = x + 1;
    until { x > 3 };
end.

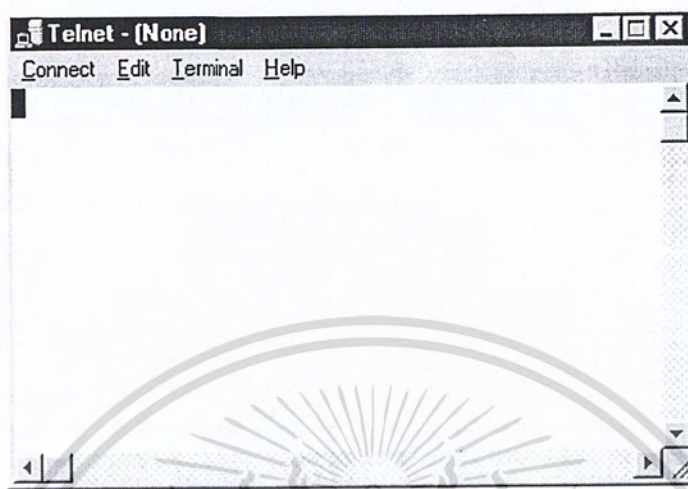
```

รูปที่ 5-9 แสดงเงื่อนไข while และ repeat-until

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.2 วิธีการสั่งงานบราวเซอร์จากระยะไกลด้วยโปรแกรมเทลเน็ต (Telnet)

1. เปิดโปรแกรมเทลเน็ต จะปรากฏหน้าต่างโปรแกรมเทลเน็ตดังรูปที่ 5.10



รูปที่ 5-10 แสดงภาพโปรแกรมเทลเน็ต

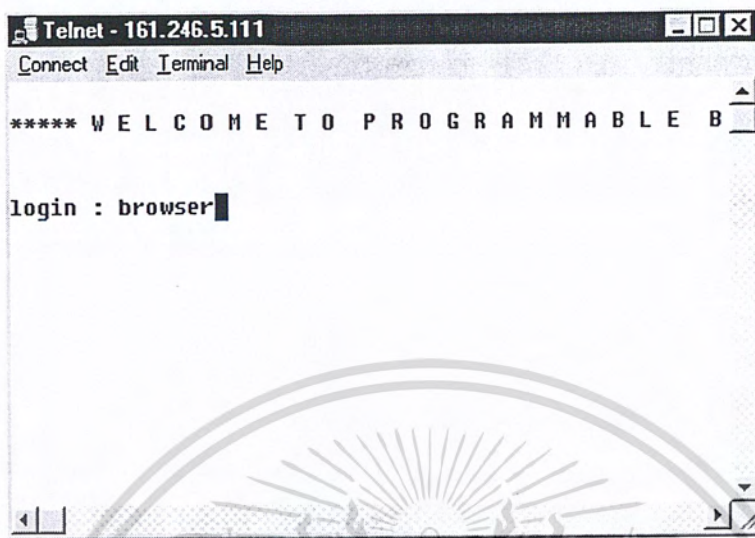
2. ระบุโฮสต์และหมายเลขพอร์ต ในที่นี้โปรแกรมใช้พอร์ตหมายเลข 6666 ดังรูปที่ 5-11



รูปที่ 5-11 แสดงการระบุชื่อโฮสต์และหมายเลขพอร์ต

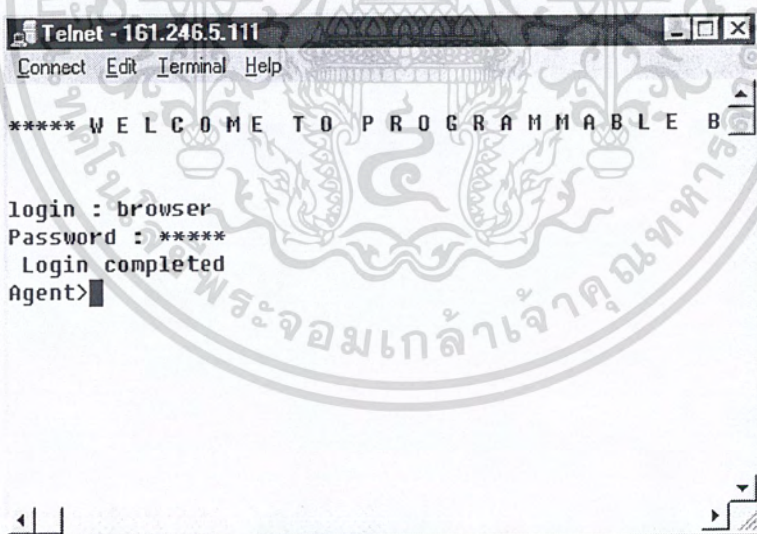
3. กดปุ่ม connect
4. จะพบกับหน้า Login เพื่อเข้าสู่ระบบ ให้ Login Name และใส่ Password ดังรูปที่ 5-12 และ รูปที่ 5-13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```
Telnet - 161.246.5.111
Connect Edit Terminal Help
***** WELCOME TO PROGRAMMABLE B
login : browser
```

รูปที่ 5-12 แสดงการกรอก Login Name



```
Telnet - 161.246.5.111
Connect Edit Terminal Help
***** WELCOME TO PROGRAMMABLE B
login : browser
Password : *****
Login completed
Agent>
```

รูปที่ 5-13 แสดงการกรอก Password

คำสั่งให้บราวเซอร์ทำงานตามสคริปต์ที่อยู่บนเครื่องโฮสต์

เมื่อต้องการให้บราวเซอร์ทำงานตามสคริปต์ที่เก็บเอาไว้ในเครื่องโฮสต์ ให้ผู้ใช้พิมพ์คำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

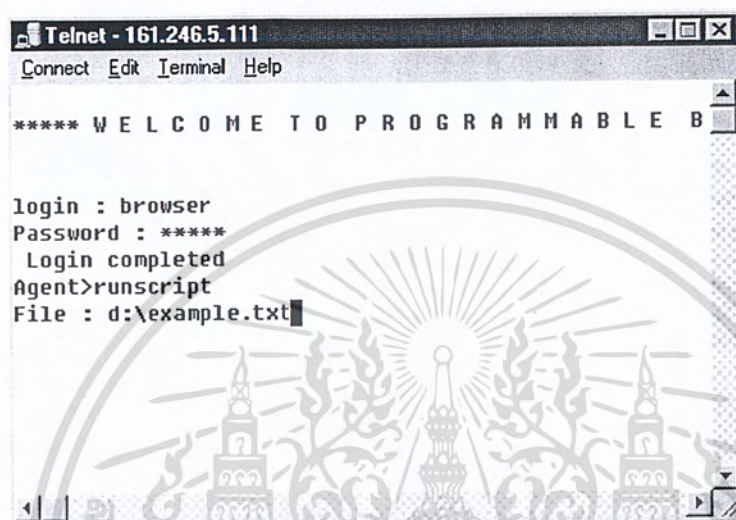
Agent>runscript

แล้วกดปุ่ม Enter

File :

แล้วระบุพาทที่เก็บสคริปต์

File : d:\example.txt



```

Telnet - 161.246.5.111
Connect Edit Terminal Help

***** WELCOME TO PROGRAMMABLE B

login : browser
Password : *****
Login completed
Agent>runscript
File : d:\example.txt
  
```

รูปที่ 5-14 แสดงคำสั่ง runscript

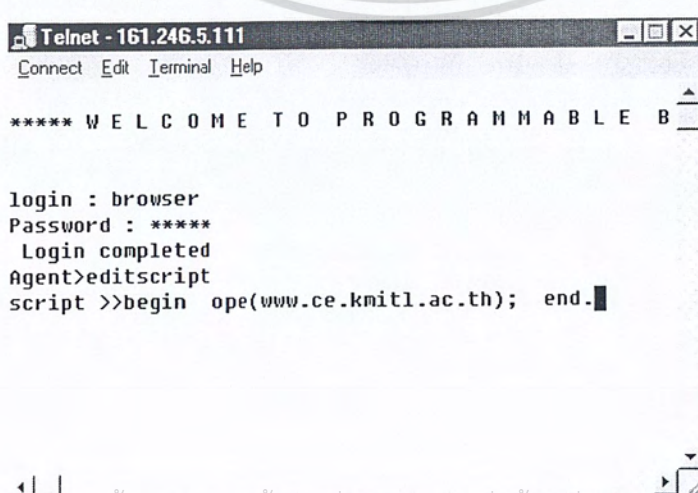
เมื่อต้องการให้บราวเซอร์ทำงานตามสคริปต์ ให้ผู้ใช้พิมพ์คำสั่ง

Agent>editscript

แล้วกด Enter จะพบกับเครื่องหมายพร้อมรับสคริปต์

script >>

จากนั้นผู้ใช้สามารถพิมพ์สคริปต์ที่ต้องการได้ เมื่อพิมพ์สคริปต์เสร็จแล้วกดปุ่ม Enter



```

Telnet - 161.246.5.111
Connect Edit Terminal Help

***** WELCOME TO PROGRAMMABLE B

login : browser
Password : *****
Login completed
Agent>editscript
script >>begin ope(www.ce.kmitl.ac.th); end.
  
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้ไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### รูปที่ 5-15 แสดงคำสั่ง editscript

เมื่อต้องการออกจากโปรแกรม ให้ผู้ใช้พิมพ์คำสั่ง

Agent>exit

แล้วกด Enter เป็นการเสร็จสิ้นการใช้งาน

```

Telnet - 161.246.5.111
Connect Edit Terminal Help

***** WELCOME TO PROGRAMMABLE B

login : browser
Password : ****
Login completed
Agent>
Agent>
Agent>exit
  
```

รูปที่ 5-16 แสดงคำสั่ง exit

### 5.3 การส่งและรับข่าวของบราวเซอร์ที่สามารถโปรแกรมได้ผ่าน NNTP

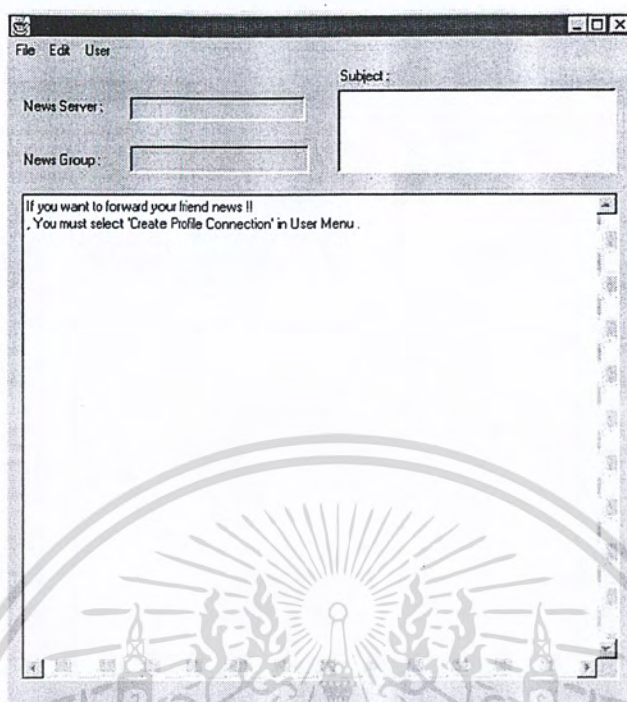
ผู้ใช้สามารถรับบริการข่าวสารจากนิวส์เซิร์ฟเวอร์ได้ โดยผ่าน โพรโตคอลเอ็นเอ็นทีพี ซึ่งโปรแกรมมี การทำงานแบ่งได้เป็น 2 กรณี คือ กรณีแรกคือ ผู้ใช้ไม่ต้องการติดต่อกับผู้ใช้คนอื่น และกรณีที่สอง ผู้ใช้ ต้องการส่งข่าวสารที่ได้รับให้แก่ผู้ใช้คนอื่นด้วย

#### 5.3.1 การรับข่าวสารจากนิวส์เซิร์ฟเวอร์

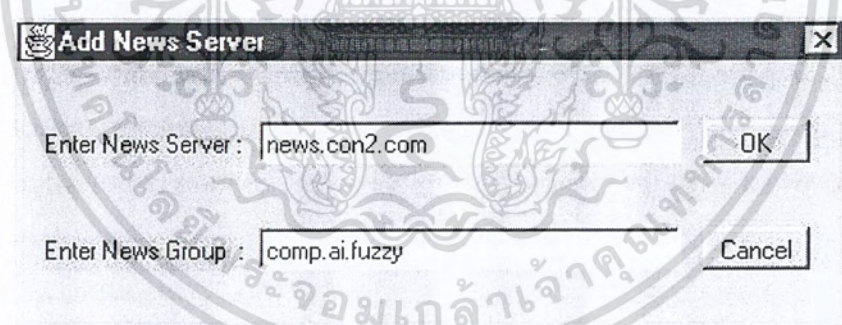
เมื่อผู้ใช้เปิดโปรแกรมเมเบิลบราวเซอร์ และเลือกที่จะรับบริการการให้ข่าวสารผ่าน โพรโตคอลเอ็นเอ็นทีพีแล้ว จะปรากฏหน้าจอ ดังรูปที่ 5-17

ผู้ใช้จะรับบริการข่าวสารได้นั้น ต้องระบุถึง นิวส์เซิร์ฟเวอร์ (News Server) ที่ต้องการดึงข่าวสารมา รวมทั้งระบุประเภทของข่าวที่สนใจ เพื่อข่าวสารที่ตรงตามความต้องการของผู้ใช้มากที่สุด ซึ่งโปรแกรมจัดการให้มีฟอร์มกรอก ดังรูปที่ 5-18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

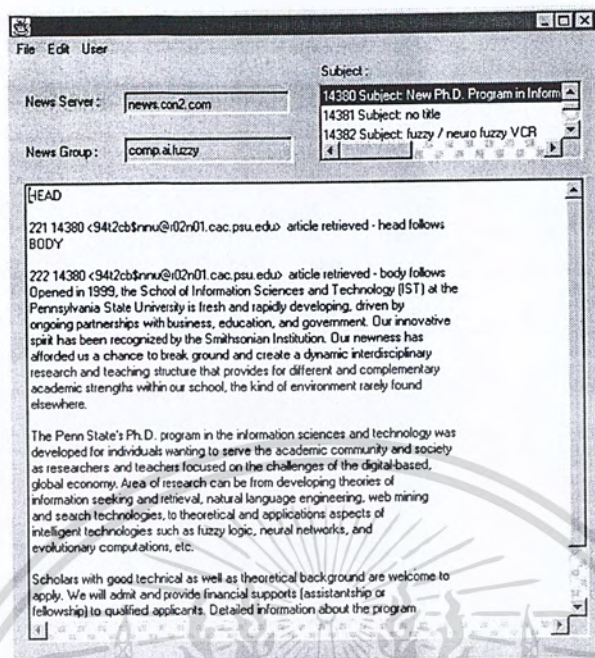


รูปที่ 5-17 แสดงหน้าต่างหลักของโปรแกรมอ่านข่าวจากนิวส์เซิร์ฟเวอร์



รูปที่ 5-18 แสดงฟอร์มรับ News Server และ News Group

เมื่อผู้ใช้เชื่อมต่อกับนิวส์เซิร์ฟเวอร์ได้แล้ว นิวส์เซิร์ฟเวอร์จะส่งข่าวสารมาให้ผู้ใช้ ซึ่งโปรแกรมได้จัดการข่าวสารที่ได้ ให้อยู่ในรูปแบบที่อ่านง่าย โดยมีการแสดงผลแบ่งเป็น 2 ส่วน คือส่วนที่แสดงหัวข้อข่าว และส่วนที่แสดงรายละเอียดของข่าว ซึ่งผู้ใช้เลือกหัวข้อข่าวที่อ่านได้ ดังรูปที่ 5-19



รูปที่ 5-19 แสดงโปรแกรมอ่านข่าวที่ไปอ่านข่าวจาก News Server แล้ว

### 5.3.2 การส่งข่าวสารที่ได้ให้แก่เอเจนต์ตัวอื่น

การส่งข่าวสารให้กับเอเจนต์ตัวอื่นจะคล้ายกับการรับข่าวสารจากนิวส์เซิร์ฟเวอร์แบบปกติ เพียงแต่มีข้อกำหนดเพิ่มเติมคือ ต้องสร้างการเชื่อมต่อไปที่เซิร์ฟเวอร์ที่ให้บริการ และต้องมียูสเซอร์โพรไฟล์แสดงสถานะของเอเจนต์ขณะนั้นด้วย ซึ่งถ้าสร้างการเชื่อมต่อไปยังเซิร์ฟเวอร์ แต่ไม่มียูสเซอร์โพรไฟล์ โปรแกรมจะแจ้งความผิดพลาด และให้ผู้ใช้กรอกฟอร์มของยูสเซอร์โพรไฟล์ ซึ่งมีลักษณะดังรูปที่ 5-20

รูปที่ 5-20 แสดงฟอร์มรับข้อมูลของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 6

### บทสรุป

#### 6.1 บทสรุปและวิจารณ์

ปฏิญญาพันธบัตรฉบับนี้แสดงให้เห็นถึงการพัฒนาโปรแกรมเว็บเบราว์เซอร์ โดยเริ่มตั้งแต่หลักการพื้นฐานต่าง ๆ ที่เกี่ยวข้อง ตลอดจนแนวคิดหลักในการพัฒนาโปรแกรม และแสดงให้เห็นถึงขั้นตอนในการพัฒนาโปรแกรม ผลลัพธ์ที่ได้ คือเบราว์เซอร์ที่สามารถโปรแกรมได้ ซึ่งมีความสามารถดังนี้

1. สามารถอ่านไฟล์เอชทีเอ็มแอลต่าง ๆ บนเครือข่ายอินเทอร์เน็ตได้
2. สามารถส่งงานด้วยภาษาสคริปต์ได้
3. สามารถทำงานได้โดยอัตโนมัติ
4. สามารถอ่านข่าวจากนิวส์เซิร์ฟเวอร์ และนิวส์กรุปที่ระบุได้
5. เบราว์เซอร์ที่สามารถโปรแกรมได้สามารถแลกเปลี่ยนข่าวสารระหว่างกันได้
6. สามารถเขียนสคริปต์เพื่อส่งข้อความเข้าเพจเจอร์โดยไม่ต้องผ่านฟอร์มของไซต์ที่ให้บริการได้
7. สามารถเขียนสคริปต์เพื่อค้นหาเว็บไซต์ที่เกี่ยวข้องกับคีย์เวิร์ดที่ผู้ใช้ระบุ โดยไม่ต้องผ่านฟอร์มของไซต์ที่ให้บริการได้
8. สามารถส่งงานได้จากกระยะไกลโดยผ่านโปรแกรมเทลเน็ต

#### 6.2 ข้อดีของเบราว์เซอร์ที่สามารถโปรแกรมได้

จากความสามารถของเบราว์เซอร์ที่สามารถโปรแกรมได้ที่กล่าวมาข้างต้น สามารถสรุปเป็นข้อดีของเบราว์เซอร์ที่สามารถโปรแกรมได้ ดังนี้

1. สามารถทำงานได้บนทุกแพลตฟอร์ม
2. สามารถทำงานได้ตามความต้องการของผู้ใช้
3. สามารถขยายเพิ่มเติมความสามารถได้อย่างไม่จำกัด

#### 6.3 ข้อเสียของเบราว์เซอร์ที่สามารถโปรแกรมได้

เนื่องจากเบราว์เซอร์ที่สามารถโปรแกรมได้นี้ ยังขาดความยืดหยุ่นในการใช้งานอยู่มาก และส่วนของภาษาสคริปต์นั้นก็ยังไม่ครอบคลุมความต้องการของผู้ใช้งานทั้งหมด ทำให้เบราว์เซอร์ที่สามารถโปรแกรมได้นี้ยังมีข้อเสียอยู่ ดังนี้

1. เบราว์เซอร์ไม่สามารถแสดงผลไฟล์บางประเภทได้ เช่น ไฟล์ ทีเอชที
2. ไม่มีการแสดงสถานะการทำงานของเบราว์เซอร์
3. โหลดเบราว์เซอร์ได้ช้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4. ไม่รองรับการทำงานของไฟล์ทรานสเฟอร์โพรโตคอล

##### 6.4 แนวทางการพัฒนา

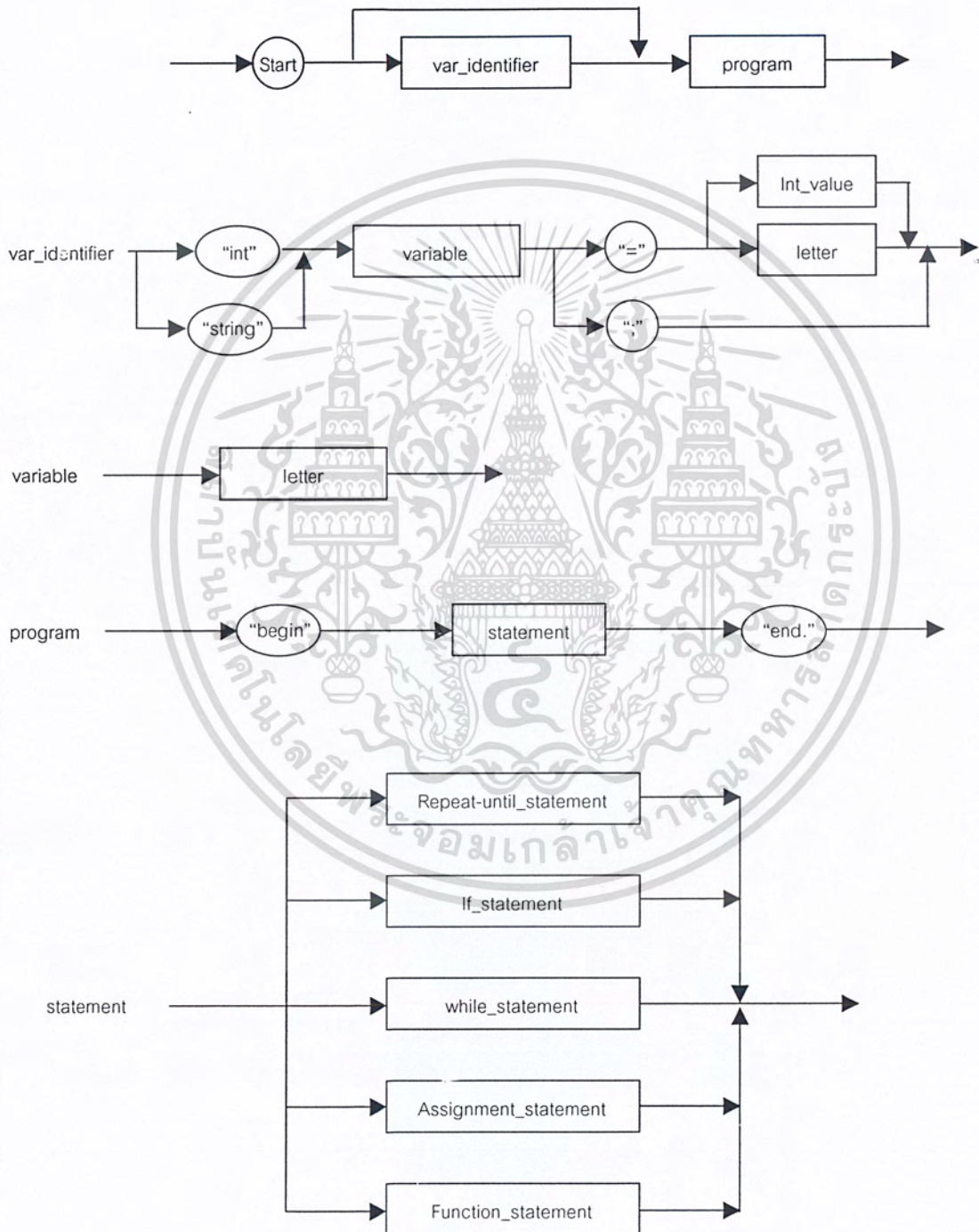
จากข้อเสียที่ได้กล่าวถึงแล้วข้างต้น จะเห็นได้ว่าโปรแกรมบราวเซอร์ที่สามารถโปรแกรมได้ยังมีสิ่งที่จะต้องพัฒนาอีกมากเพื่อให้ได้โปรแกรมที่มีความสมบูรณ์และตรงกับความต้องการของผู้ใช้มากที่สุด สิ่งสมควรได้รับการพัฒนามี ดังนี้

1. เพิ่มความสามารถของภาษาสคริปต์
2. พัฒนابราวเซอร์ให้มีความสามารถเหมือนกับบราวเซอร์ทั่วไป
3. พัฒนابราวเซอร์ที่สามารถโปรแกรมได้ให้ชาญฉลาด

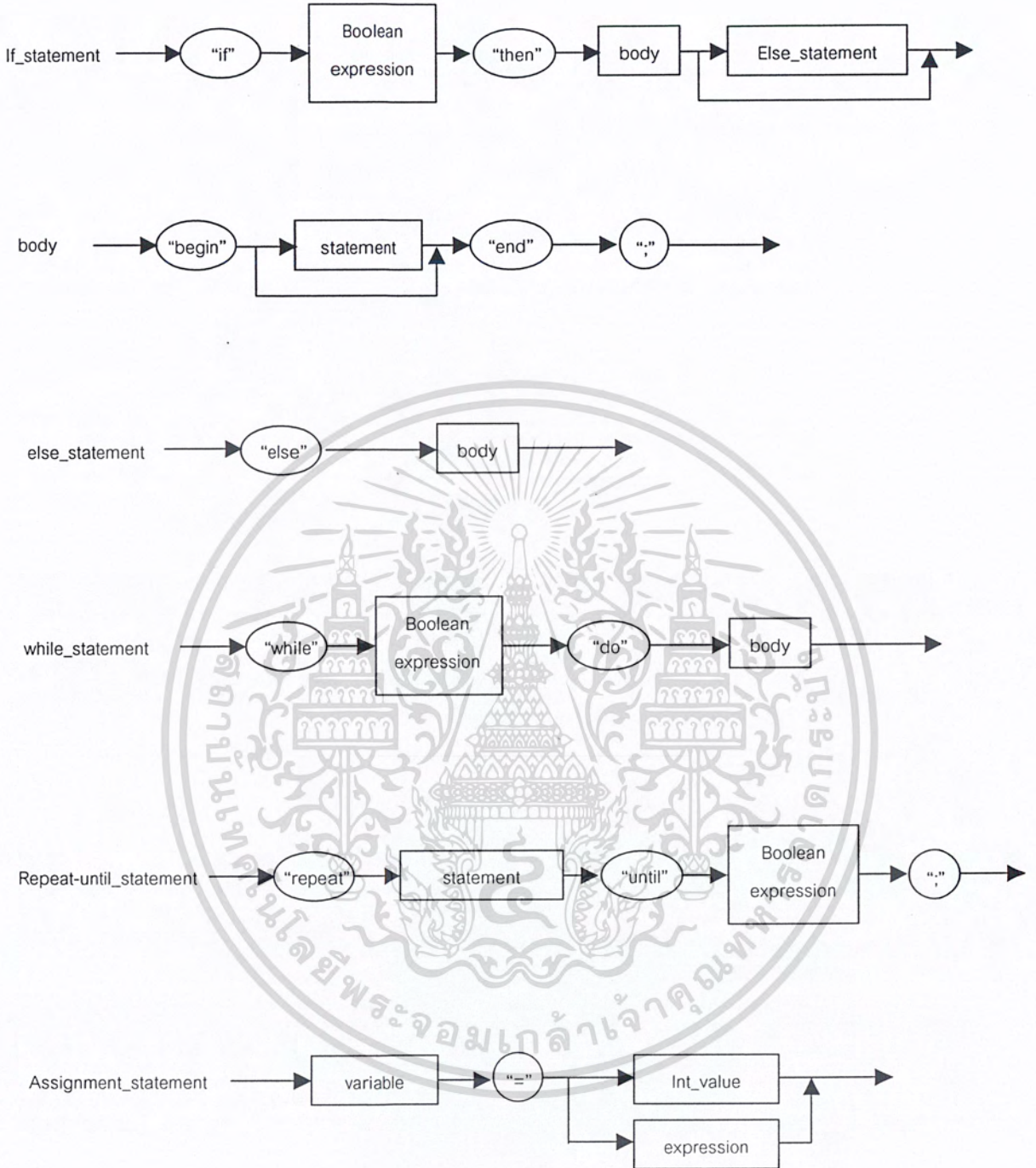


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

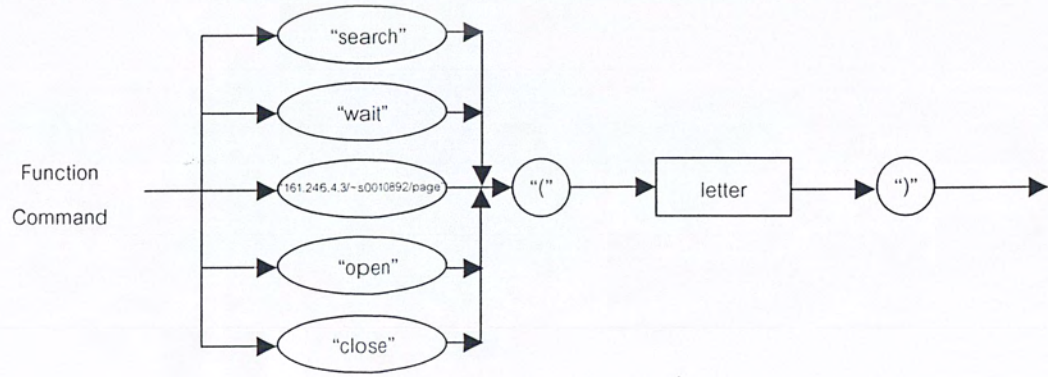
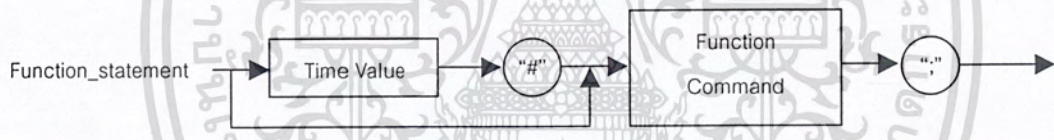
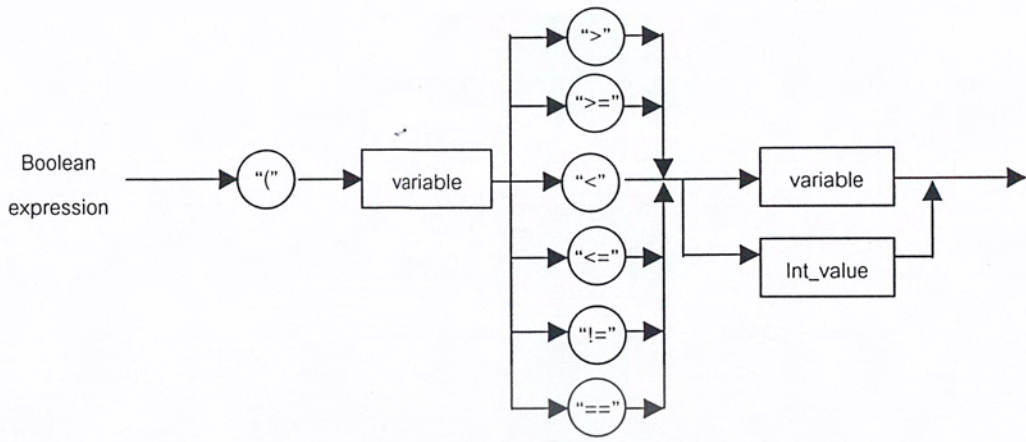
## ภาคผนวก ก. BNF ของภาษาสคริปต์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ข.

## ตัวอย่าง Public News Server

News server hostname <sup>2</sup>	Posting <sup>2</sup>	Group s <sup>2</sup>	Speed <sup>2</sup>	Average article retention <sup>2</sup>	Added <sup>?</sup>
<a href="#">alex.unibel.by</a> (info)	yes	20	0.15k/s ec	297 (5946 reported articles in 20 groups)	2001-04-15 02:36:26
<a href="#">ns.ukrcom.net.ua</a> (info)	yes	21	0.41k/s ec	4004 (84086 reported articles in 21 groups)	2001-04-09 00:20:26
<a href="#">ala.netbeans.com</a> (info)	yes	3	0.21k/s ec	3080 (9242 reported articles in 3 groups)	2001-04-07 10:19:07
<a href="#">206.79.157.88</a> (info)	yes	64	2.44k/s ec	6471 (414158 reported articles in 64 groups)	2001-04-07 07:47:07
<a href="#">www.trumpet.com.au</a> (info)	yes	8	0.16k/s ec	952 (7616 reported articles in 8 groups)	2001-04-07 07:22:28
<a href="#">sigmadesigns.com</a> (info)	yes	5	0.51k/s ec	1241 (6206 reported articles in 5 groups)	2001-04-07 04:56:55
<a href="#">www.datadynamics.com</a> (info)	yes	31	1.02k/s ec	355 (11023 reported articles in 31 groups)	2001-04-07 04:06:44
<a href="#">netforum.netvigator.com</a> (info)	yes	85	2.10k/s ec	927 (78819 reported articles in 85 groups)	2001-04-07 02:28:46
<a href="#">143.93.54.1</a> (info)	yes	14	0.34k/s ec	25 (363 reported articles in 14 groups)	2001-04-07 01:00:31
<a href="#">autodesk.webcrossing.com</a> (info)	yes	78	3.56k/s ec	7067 (551296 reported articles in 78 groups)	2001-04-07 00:41:59
<a href="#">news.vci.com</a> (info)	yes	2	0.02k/s ec	546 (1093 reported articles in 2 groups)	2001-04-06 22:27:50
<a href="#">user81.lokigames.com</a> (info)	yes	31	1.57k/s ec	850 (26379 reported articles in 31 groups)	2001-04-06 22:02:17
<a href="#">e-post.inode.at</a> (info)	yes	1	0.02k/s	18 (18 reported articles in 1 groups)	2001-04-06

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			ec		20:54:24
<a href="#">rgsoftware.com</a> (info)	yes	5	0.25k/s ec	6 (32 reported articles in 5 groups)	2001-04-06 11:40:19
<a href="#">195.211.161.174</a> (info)	yes	8	0.60k/s ec	35 (285 reported articles in 8 groups)	2001-04-06 11:30:39
<a href="#">204.198.135.111</a> (info)	yes	5	0.32k/s ec	43 (218 reported articles in 5 groups)	2001-04-06 02:40:55
<a href="#">h04.sny.collab.net</a> (info)	yes	9	0.43k/s ec	1795 (16162 reported articles in 9 groups)	2001-04-06 00:12:28
<a href="#">webdiscuss.com</a> (info)	yes	8	0.63k/s ec	56 (455 reported articles in 8 groups)	2001-04-05 23:11:26
<a href="#">h-204-29-187-152.netscape.com</a> (info)	yes	131	13.39k/ sec	2214 (290113 reported articles in 131 groups)	2001-04-05 23:10:20
<a href="#">masamune.dais.is.tohoku.ac.jp</a> (info)	no	1321	21.39k/ sec	54 (72653 reported articles in 1321 groups)	2001-04-05 07:28:44
<a href="#">news.boulder.ibm.com</a> (info)	yes	154	4.21k/s ec	1033 (159219 reported articles in 154 groups)	2001-04-05 05:00:26
<a href="#">www.starkey.com</a> (info)	no	67	3.09k/s ec	8509 (570115 reported articles in 67 groups)	2001-03-28 19:33:42
<a href="#">news.gamma.ru</a> (info)	yes	3078	16.53k/ sec	134 (412945 reported articles in 3078 groups)	2001-03-27 05:00:49
<a href="#">news.baku.or.jp</a> (info)	yes	40	1.18k/s ec	1704 (68195 reported articles in 40 groups)	2001-03-26 11:57:04
<a href="#">204.213.191.229</a> (info)	yes	29	0.81k/s ec	3851 (111695 reported articles in 29 groups)	2001-03-24 12:05:41
<a href="#">xcom.pl</a> (info)	yes	18	0.67k/s ec	236 (4260 reported articles in 18 groups)	2001-03-24 00:07:02
<a href="#">64.14.86.166</a> (info)	no	3642	35.03k/ sec	280 (1022308 reported articles in 3642 groups)	2001-03-22 23:48:22
<a href="#">ticperm.ozemail.com.au</a> (info)	yes	41	0.70k/s	0 (2 reported articles in 41 groups)	2001-03-22 21:50:54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			ec		
<a href="#">212.125.94.148</a> (info)	yes	38912	21.38k/ sec	40 (1566849 reported articles in 38912 groups)	2001-03-22 21:45:55
<a href="#">stalker2.alfacom.net</a> (info)	yes	3	0.01k/s ec	1461 (4384 reported articles in 3 groups)	2001-03-22 14:12:15
<a href="#">web.enterprisebeans.de</a> (info)	yes	2	0.24k/s ec	18 (36 reported articles in 2 groups)	2001-03-22 11:35:01
<a href="#">24.69.160.66.on.wave.home.com</a> (info)	yes	30965	18.04k/ sec	0 (16524 reported articles in 30965 groups)	2001-03-22 02:48:12
<a href="#">adsl-213-254-163-44.mistral-uk.net</a> (info)	yes	1	0.02k/s ec	8 (8 reported articles in 1 groups)	2001-03-21 21:38:09
<a href="#">cherry.rt.ru</a> (info)	yes	2782	10.95k/ sec	5608 (15603709 reported articles in 2782 groups)	2001-03-19 11:48:17
<a href="#">server.arg.sulinet.hu</a> (info)	yes	3127	5.92k/s ec	1 (5315 reported articles in 3127 groups)	2001-03-19 08:16:17
<a href="#">narzisse.hrz.tfh-wildau.de</a> (info)	no	16127	14.01k/ sec	113 (1833181 reported articles in 16127 groups)	2001-03-19 02:12:11
<a href="#">news.epita.fr</a> (info)	no	3777	10.26k/ sec	255 (965889 reported articles in 3777 groups)	2001-03-18 22:55:35
<a href="#">hawking.pct.edu</a> (info)	yes	1616	18.50k/ sec	472 (764047 reported articles in 1616 groups)	2001-03-18 19:09:58
<a href="#">news2.demos.su</a> (info)	no	12670	21.55k/ sec	2180 (27632946 reported articles in 12670 groups)	2001-03-18 12:38:30
<a href="#">ueplac.kiev.ua</a> (info)	yes	8	0.07k/s ec	1 (11 reported articles in 8 groups)	2001-03-12 21:49:43
<a href="#">loke.ammulti.dk</a> (info)	yes	1	0.09k/s ec	154 (154 reported articles in 1 groups)	2001-03-12 11:11:58
<a href="#">brainiac.rz.uni-ulm.de</a> (info)	yes	12	0.40k/s ec	1169 (14028 reported articles in 12 groups)	2001-03-12 04:25:51
<a href="#">c217495-a.peoria1.il.home.com</a> (info)	yes	2	0.17k/s	4 (8 reported articles in 2 groups)	2001-03-12 02:38:06

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			ec		
<a href="#">203.92.80.163</a> (info)	yes	46	1.01k/s ec	2343 (107823 reported articles in 46 groups)	2001-03-12 01:07:09
<a href="#">c646697-a.eugene1.or.home.com</a> (info)	yes	38	1.23k/s ec	9 (365 reported articles in 38 groups)	2001-03-12 00:08:08
<a href="#">216.0.165.219</a> (info)	yes	5	0.10k/s ec	1173 (5865 reported articles in 5 groups)	2001-03-11 20:49:52
<a href="#">62.210.73.240</a> (info)	yes	444	5.72k/s ec	1664 (739127 reported articles in 444 groups)	2001-03-11 19:13:02
<a href="#">news.internetia.com</a> (info)	yes	15	0.56k/s ec	17 (261 reported articles in 15 groups)	2001-03-11 16:20:06
<a href="#">helena.kollegienet.dk</a> (info)	yes	21	0.17k/s ec	212 (4453 reported articles in 21 groups)	2001-03-11 10:32:02
<a href="#">mimer.miba.auc.dk</a> (info)	yes	73	2.95k/s ec	8604 (628117 reported articles in 73 groups)	2001-03-11 06:41:12
<a href="#">infosun2.rus.uni-stuttgart.de</a> (info)	no	26216	38.79k/ sec	780 (20457189 reported articles in 26216 groups)	2001-03-08 15:26:13
<a href="#">www.dedenet.de</a> (info)	yes	22	0.71k/s ec	466 (10262 reported articles in 22 groups)	2001-03-07 10:59:20
<a href="#">cnews1.corel.ca</a> (info)	yes	295	12.71k/ sec	1227 (362207 reported articles in 295 groups)	2001-03-05 01:26:18
<a href="#">asplists.lyris.net</a> (info)	yes	441	0.93k/s ec	127159 (56077444 reported articles in 441 groups)	2001-03-05 01:08:15
<a href="#">milzis.latnet.lv</a> (info)	yes	11	0.14k/s ec	289 (3188 reported articles in 11 groups)	2001-03-05 00:42:36
<a href="#">chaos052.greynware.com</a> (info)	yes	544	3.97k/s ec	187 (102025 reported articles in 544 groups)	2001-03-05 00:26:38
<a href="#">216.18.96.240</a> (info)	yes	16	0.64k/s ec	493 (7903 reported articles in 16 groups)	2001-03-05 00:14:45
<a href="#">www.starkey.com</a> (info)	no	67	3.09k/s	8506 (569918 reported articles in 67 groups)	2001-03-05 00:02:06

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			ec		
<a href="http://starnews.sun.com">starnews.sun.com</a> (info)	yes	48	3.65k/s ec	37 (1822 reported articles in 48 groups)	2001-03-04 23:52:04
<a href="http://www.anokaramsey.mnscu.edu">www.anokaramsey.mnscu.edu</a> (info)	yes	1	0.09k/s ec	9 (9 reported articles in 1 groups)	2001-03-04 23:41:32
<a href="http://masamune.dais.is.tohoku.ac.jp">masamune.dais.is.tohoku.ac.jp</a> (info)	no	1321	25.67k/ sec	54 (72311 reported articles in 1321 groups)	2001-03-02 11:18:53
<a href="http://mailhost.szut.uni-bremen.de">mailhost.szut.uni-bremen.de</a> (info)	yes	7173	5.90k/s ec	149 (1068804 reported articles in 7173 groups)	2001-03-02 11:09:17
<a href="http://news.softvelocity.com">news.softvelocity.com</a> (info)	yes	54	0.50k/s ec	1795 (96942 reported articles in 54 groups)	2001-02-28 08:18:58
<a href="http://epigone.morat.net">epigone.morat.net</a> (info)	yes	64	2.03k/s ec	2 (168 reported articles in 64 groups)	2001-02-28 02:11:49
<a href="http://news2.ipartners.pl">news2.ipartners.pl</a> (info)	no	220	7.40k/s ec	3858 (848829 reported articles in 220 groups)	2001-02-19 19:31:13
<a href="http://www.tmbg.org">www.tmbg.org</a> (info)	yes	2	0.24k/s ec	856 (1712 reported articles in 2 groups)	2001-02-18 13:53:02
<a href="http://bar.winnisurf.de">bar.winnisurf.de</a> (info)	yes	87	2.95k/s ec	5 (462 reported articles in 87 groups)	2001-02-18 13:43:40
<a href="http://isttmoj0.customer.nettuno.it">isttmoj0.customer.nettuno.it</a> (info)	yes	26	0.45k/s ec	4 (111 reported articles in 26 groups)	2001-02-18 07:28:07
<a href="http://194.98.205.2">194.98.205.2</a> (info)	yes	154	2.80k/s ec	1033 (159215 reported articles in 154 groups)	2001-02-17 22:11:43
<a href="http://mail.opera.no">mail.opera.no</a> (info)	yes	38	0.86k/s ec	2535 (96347 reported articles in 38 groups)	2001-02-17 21:10:33
<a href="http://news.univ-nantes.fr">news.univ-nantes.fr</a> (info)	yes	3199	13.99k/ sec	12515 (40038394 reported articles in 3199 groups)	2001-02-17 15:11:30
<a href="http://cogito.univ.gda.pl">cogito.univ.gda.pl</a> (info)	yes	5	0.03k/s ec	42 (210 reported articles in 5 groups)	2001-02-17 12:43:38
<a href="http://nntp.tfvs.tp.edu.tw">nntp.tfvs.tp.edu.tw</a> (info)	yes	15377	4.49k/s	636 (9783329 reported articles in 15377 groups)	2001-02-17 08:54:33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			ec		
<a href="#">212.33.64.94</a> (info)	yes	1	0.02k/s ec	2569 (2569 reported articles in 1 groups)	2001-02-16 22:46:09
<a href="#">cvg-65-26-144-21.cinci.rr.com</a> (info)	yes	2	0.36k/s ec	28 (56 reported articles in 2 groups)	2001-02-16 20:12:52
<a href="#">aspen.earthnet.net</a> (info)	yes	4	0.05k/s ec	1 (4 reported articles in 4 groups)	2001-02-16 12:56:12
<a href="#">73.64.242.195.infosources.fr</a> (info)	no	33313	29.85k/ sec	62 (2095274 reported articles in 33313 groups)	2001-02-16 09:51:16
<a href="#">it.pol.dk</a> (info)	yes	9	0.37k/s ec	437 (3939 reported articles in 9 groups)	2001-02-16 08:00:59
<a href="#">ussenterprise.ufp.org</a> (info)	yes	2	0.49k/s ec	852 (1705 reported articles in 2 groups)	2001-02-16 04:19:08

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

- [1] ดร.วีระศักดิ์ ชิงदार : “*Fundamental of Java Programming Volume I* ” , SUM System Company Limited, 2541.
- [2] ทรงเกียรติ ภาวดี : “*แกะรอย CGI เพื่อเขียนสคริปต์เรียกเพจเจอร์/มือถือผ่านเว็บ*”, บริษัท วิตดี กรุ๊ป จำกัด, 2537.
- [3] กิตติ ภัคดีวัฒนะกุล : “*JAVA ฉบับโปรแกรมเมอร์*” , บริษัท เคทีพี คอมพ์ แอนด์ คอนซัลท์ จำกัด, 2542.
- [4] Bill McCarty and Luke Cassidy-Dorion : “*JAVA Distributed Objects*”, Sams, 1999.
- [5] Joseph P. Bigus and Jennifer Bigus : “*Constructing Intelligent Agents with Java*”, Wiley Computer Publishing, 1997.
- [6] Merlin Hughes, Michael Shoffner and Derek Hamner : “*Java Network Programming*”, Manning Publications Co., 1999.
- [7] Danny B. Lange, Mitsuru Oshima : “*Programming and Deploying Java Mobile Agents with Aglets*”, Addison Wesley Longman, Inc., 1998.
- [8] Mark Watson : “*Intelligent Java Application for the Internet and Intranets*”, Morgan Kaufmann Publishers, 1997.
- [9] “*Webopedia: Online Computer Dictionary for Internet Terms and Technical Support*”, <http://www.pcwebopedia.com>
- [10] “*Intelligent Software Agents on the Internet*”, <http://www.hermans.org/agents/index.html>
- [11] “*Evaluation of intelligent software agents*”, <http://w3.informatik.gu.se/~dixi/agent/agent.htm>
- [12] “*UMBC Agent Web*”, <http://agents.umbc.edu/>
- [13] “*Public News Servers sorted by longevity*”, <http://www.newzbot.com/sorted-longevity-table.html>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้