

บทที่ 2

ทฤษฎีที่เกี่ยวข้องกับงานวิจัย

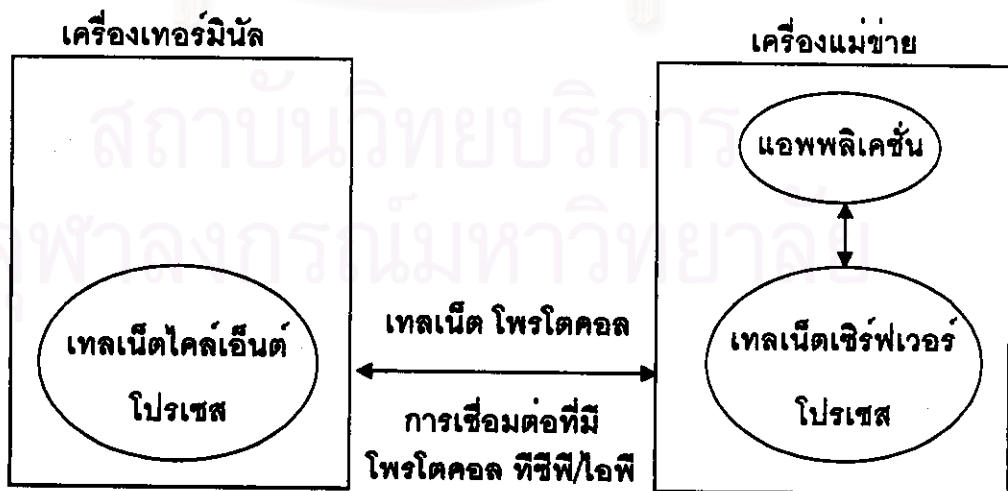
หลักการของการเจรจาตัวเลือกของเทลเน็ต

1. เทลเน็ตโพรโตคอล (Telnet Protocol)

เทลเน็ตโพรโตคอล จะเป็นรูปแบบของการติดต่อสื่อสารในระบบเครือข่าย โดยจุดประสงค์ เพื่อจะเป็นมาตรฐานในการกำหนดวิธีการติดต่อของอุปกรณ์เทอร์มินัล กับเครื่องแม่ข่ายต่าง ๆ ในระบบเครือข่าย โดยมีลักษณะการส่งข้อมูลขนาด 8 บิต แบบสองทิศทาง (bi - directional)

2. ตัวแบบของเทลเน็ต (Telnet Model)

ผู้ใช้เทอร์มินัลจะทำการติดต่อกับเทลเน็ตไคลเอ็นต์โพรเซส (Telnet Client Process) ในเทอร์มินัลที่ใช้งานอยู่ และเทลเน็ตไคลเอ็นต์ (Telnet Client) จะทำการแลกเปลี่ยนข้อมูลกับเทลเน็ตเซิร์ฟเวอร์ (Telnet Server) บนการเชื่อมต่อที่ใช้โพรโตคอล ทีซีพี/ไอพี และเทลเน็ตเซิร์ฟเวอร์ติดต่อกับแอปพลิเคชัน โดยเลียนแบบการทำงานเป็นเทอร์มินัลของเซิร์ฟเวอร์ (Server) ซึ่งสามารถแสดงได้ดังรูปที่ 2.1



รูปที่ 2.1 แสดงการติดต่อระหว่าง เทลเน็ตไคลเอ็นต์ กับเทลเน็ตเซิร์ฟเวอร์

เทลเน็ตโพรโตคอล ถูกสร้างโดยอาศัย หลักการ 3 ประการ (Postel, 1983) คือ

- ก. แนวความคิดเกี่ยวกับเทอร์มินัลเสมือน
- ข. พื้นฐานของการเจรจาตัวเลือก
- ค. หลักการของความสมมาตรของเทอร์มินัลและโปรเซส

ก. แนวความคิดเกี่ยวกับเทอร์มินัลเสมือน

เมื่อมีการติดต่อโดยใช้เทลเน็ต แต่ละฝั่งเป็นเทอร์มินัลเสมือน ซึ่งเป็นอุปกรณ์สื่อสารอักขระสองทิศทาง (bi - direction character device) สำหรับโต้ตอบกับผู้ใช้ ซึ่งประกอบด้วย 2 ส่วน คือ ส่วนรับข้อมูล คือ แป้นพิมพ์ และส่วนแสดงผล ซึ่งอาจเป็นจอภาพ หรือเครื่องพิมพ์ มิได้แสดงข้อมูลที่เข้ามาจากเครือข่ายหรือข้อมูลจากแป้นพิมพ์ ส่วนข้อมูลที่ต้องการส่งไปในเครือข่ายเป็นตามอักขระรหัสแอสกี (ASCII code) ขนาด 8 บิต โดยปกติบิตที่ 8 จะมีค่าเป็น 0

ข. พื้นฐานของการเจรจาตัวเลือก

เทลเน็ตโพรโตคอล มีส่วนที่กำหนดข้อตกลง ในการติดต่อกันระหว่างเครื่องบนระบบเครือข่าย โดยใช้รหัสขอการรับบริการ (request code) และรหัสตัวเลือก (option code) ซึ่งถูกบรรจุไว้ในข้อมูลที่ส่งติดต่อกัน

โดยพื้นฐานของการกำหนดการใช้ตัวเลือก ระหว่างเทลเน็ตไคลเอ็นต์กับเทลเน็ตเซิร์ฟเวอร์ จะมีฝั่งหนึ่งทำการส่งรหัสขอการรับบริการ ด้วยการส่งรหัสตัวเลือกไปหา ส่วนอีกฝั่งจะยอมรับหรือปฏิเสธการทำงานตามรหัสตัวเลือกที่ได้รับมาก็ได้ รหัสตัวเลือกของการเจรจาตัวเลือกมีอยู่มากมาย แต่ในการพัฒนาโปรแกรมในงานวิจัยนี้ไม่จำเป็นต้องใช้ทุกรหัสตัวเลือก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ความหมายของรหัสขอการรับบริการ และรหัสตัวเลือกที่สำคัญ สามารถแสดงได้
ดังตารางที่ 2.1 และตารางที่ 2.2

รหัสขอการรับบริการ	คำอธิบาย	ค่ารหัสฐานสิบ
WILL	ยินดีทำตาม รหัสตัวเลือกที่ส่งไป	251
WONT	ไม่ยินดีทำตาม รหัสตัวเลือกที่ได้รับ	252
DO	ต้องการให้ทำตาม รหัสตัวเลือกที่ส่งไป	253
DONT	ไม่ต้องการให้ทำตาม รหัสตัวเลือกที่ได้รับ	254
SB (Begin of Subnegotiation)	เริ่มต้นเข้าสู่การเจรจาย่อย	250
SE (End of Subnegotiation)	ออกจากกรทำการเจรจาย่อย	240

ตารางที่ 2.1 แสดงความหมายของรหัสขอการรับบริการ

รหัสตัวเลือก	คำอธิบาย	ค่ารหัสฐานสิบ
BINARY	เริ่มการส่งข้อมูลไบนารี ขนาด 8 บิต	0
ECHO	แสดงอักขระไปยังหน่วยแสดงผลของ ไคล์เอ็นต์	1
SGA (Suppress Go Ahead)	ขอทำการติดต่อแบบ 2 ทิศทาง	3
TERMTYPE	ชนิดของเทอร์มินัลที่เซิร์ฟเวอร์รู้จักและ ไคล์เอ็นต์สามารถเลียนแบบ	24
NAWS	ข้อมูลความกว้างและความสูงของจอ ภาพที่ใช้บนฝั่งไคล์เอ็นต์	31

ตารางที่ 2.2 แสดงความหมายของรหัสตัวเลือกที่สำคัญ

สำหรับรหัสขอการรับบริการในการทำการเจรจาตัวเลือก มีด้วยกัน 4 รูปแบบ

- | | |
|-----------------------|--|
| 1. DO (option code) | ขอให้กระทำตามรหัสตัวเลือก |
| WILL (option code) | คู่ติดต่อเห็นด้วย ยินดีทำตามรหัสตัวเลือก |
| 2. DO (option code) | ขอให้กระทำตามรหัสตัวเลือก |
| WONT (option code) | คู่ติดต่อปฏิเสธ และไม่เปลี่ยนสถานะ |
| 3. WILL (option code) | เป็นการบอกว่ายินดีทำตามรหัสตัวเลือก |
| DO (option code) | คู่ติดต่อจะให้ทำตามรหัสตัวเลือก |
| 4. WILL (option code) | เป็นการบอกว่ายินดีทำตามรหัสตัวเลือก |
| DONT (option code) | คู่ติดต่อปฏิเสธ และไม่เปลี่ยนสถานะ |
| WONT (option code) | เป็นการย้ำว่าจะไม่เปลี่ยนสถานะ |

ตัวอย่างรูปแบบของการติดต่อเช่น ฝั่งหนึ่งมีการส่ง WILL ECHO จะมีความหมายว่า ยินดีทำตามรหัสตัวเลือก ECHO โดยอีกฝั่งสามารถส่ง DO ECHO เพื่อเป็นการตอบรับในลักษณะต้องการให้ทำ หรือ DONT ECHO ในกรณีที่ไม่ต้องการให้ทำ ถ้าในกรณีที่ไม่ต้องการให้ทำตามรหัสตัวเลือก ECHO หลังจากที่ฝั่งที่ยินดีทำตามรหัสตัวเลือก ECHO ได้รับ DONT ECHO แล้ว จะทำการส่ง WONT ECHO เพื่อเป็นการย้ำว่าไม่มีการทำงานใด ๆ ตามรหัสตัวเลือกนั้น

ในลักษณะเดียวกันเมื่อมีการส่ง DO ECHO มีความหมายว่า ขอให้ฝั่งตรงข้ามทำตามตัวเลือก ECHO โดยอีกฝั่งสามารถส่ง WILL ECHO เพื่อเป็นการตอบรับในลักษณะยินดีทำตาม หรือ WONT ECHO ในกรณีที่ไม่ยินดีทำตาม

เมื่อมีการส่งข้อมูลของการเจรจาตัวเลือก ต้องมีไบต์แรกเป็นรหัสคำสั่ง (Interpret As Command) ซึ่งแทนรหัสด้วย IAC มีค่ารหัสฐานสิบ คือ 255 ข้อมูลของการเจรจาตัวเลือกที่จะส่ง โดยปกติประกอบด้วยข้อมูลขนาด 3 ไบต์ ซึ่งประกอบด้วย รหัสคำสั่ง รหัสขอการรับบริการ และ รหัสตัวเลือก ตัวอย่างเช่น ต้องการส่ง WILL TERMTYPE ข้อมูลที่ส่ง คือ

IAC WILL TERMTYPE

แต่ในกรณีที่มีการทำการเจรจาตัวเลือกย่อย (subnegotiation) จะมีการส่งข้อมูลมากกว่า 3 ไบต์ ตัวอย่างเช่น เทลเน็ตเซิร์ฟเวอร์ต้องการให้ส่งชนิดของเทอร์มินัลที่ต้องการติดต่อด้วย จากนั้น เทลเน็ตไคลเอ็นต์ทำการแจ้งว่าต้องการติดต่อด้วยเทอร์มินัลชนิด VT100 ซึ่งจะมีรูปแบบการเจรจาตัวเลือกย่อยดังนี้

เซิร์ฟเวอร์ : IAC SB TERMTYPE SEND IAC SE

ไคลเอ็นต์ : IAC SB TERMTYPE IS "VT100" IAC SE

ค. หลักการของความสมมาตรของเทอร์มินัลและโปรเซส

ในการเจรจาตัวเลือกระหว่างเทลเน็ตไคลเอ็นต์ กับเทลเน็ตเซิร์ฟเวอร์ คู่ติดต่อจะมองรหัสตัวเลือกที่รับมาเป็นเหมือนคำสั่ง ซึ่งต้องตอบกลับ ถ้าคู่ติดต่อได้รับการร้องขอการรับบริการ ซึ่งเป็นไปตามสถานะของการทำงานเดิมอยู่แล้ว ไม่ควรตอบกลับ เพื่อป้องกันการทำงานวนซ้ำแบบไม่สิ้นสุด (infinite loop) ดังนั้นในการทำการเจรจาตัวเลือก จะตอบกลับเฉพาะเมื่อมีการร้องขอการรับบริการ สำหรับเปลี่ยนสถานะการทำงานเท่านั้น

ข้อแตกต่างระหว่างการทำงานภายใต้ตอสกับภายใต้วินโดว

การทำงานในตอสนั้น ไม่มีระบบจัดการปันส่วนทรัพยากรใด ๆ เมื่อแอฟพลิเคชันหนึ่ง ๆ ทำงาน จะถือเสมือนว่าแอฟพลิเคชันนั้น เป็นเจ้าของทรัพยากรทั้งหมด เช่น แป้นพิมพ์ จอภาพ หน่วยความจำ หรือแม้กระทั่งซีพียู แต่สำหรับวินโดวจะมีการบังคับให้ แอฟพลิเคชันมีการปันส่วนทรัพยากรต่าง ๆ ใช้ร่วมกันเสมอ

ในวินโดวเป็นระบบหลายงาน (multitasking) พื้นที่หน่วยความจำซึ่งเป็นทรัพยากรจำกัด จะต้องปันส่วนกันใช้ เพื่อให้แอฟพลิเคชันต่างทำงานพร้อมกันได้ ในการทำงานของวินโดว จะมีการเคลื่อนย้ายสลับเปลี่ยน (swap) หรือทิ้ง (discard) ส่วนต่าง ๆ ของหน่วยความจำ ตลอดเวลา เพื่อให้ทำงานทั้งระบบมีประสิทธิภาพสูงสุด การใช้หน่วยความจำแต่ละครั้ง พบว่ามีอยู่หลายครั้งที่ข้อมูลที่ใส่เข้าไป ไม่อยู่ตำแหน่งเดิม

การพัฒนาโปรแกรมคอสเพื่อใช้งานภายใต้เอ็มเอสดอสวินโดว์ของไมโครซอฟต์วินโดว์

เอ็มเอสดอสวินโดว์ เป็นโปรแกรมที่อยู่ภายใต้ไมโครซอฟต์วินโดว์ เพื่อให้ผู้ใช้สามารถทำงานแอฟพลิเคชันสำหรับดอส โดยไม่ต้องออกจากไมโครซอฟต์วินโดว์ แต่แอฟพลิเคชันสำหรับดอสบางตัวไม่สามารถทำงานภายใต้เอ็มเอสดอสวินโดว์ได้ อาทิ โปรแกรมดิสก์ออปติไมซ์ โปรแกรมจัดเรียงฮาร์ดดิสก์ เป็นต้น ควรใช้โปรแกรมเหล่านี้ที่เป็นเวอร์ชันสำหรับวินโดว์

แอฟพลิเคชันดอสที่ทำงานเกี่ยวกับการติดต่อทางเครือข่าย เพื่อใช้งานภายใต้เอ็มเอสดอสวินโดว์ของไมโครซอฟต์วินโดว์ ต้องระมัดระวังเมื่อผู้ใช้เปลี่ยนไปใช้งานอื่น (task switching) โดยการกดแป้นพิมพ์ ALT - Tab เพราะจะทำให้การติดต่อระหว่างเครื่องผ่านทางเครือข่ายขาดการติดต่อกันได้ เนื่องจากเป็นแอฟพลิเคชันดอส ซึ่งไม่มีระบบจัดการปันส่วนทรัพยากร เมื่อแอฟพลิเคชันดอสใด ๆ ทำงาน จะถือเสมือนว่าแอฟพลิเคชันนั้น เป็นเจ้าของทรัพยากรทั้งหมด เช่น แป้นพิมพ์ จอภาพ หน่วยความจำ เน็ตเวิร์คอินเตอร์เฟสการ์ด หรือแม้กระทั่งซีพียู

ดังนั้นเพื่อป้องกันการแย่งกันใช้ทรัพยากรต่าง ๆ ขณะที่ใช้งานแอฟพลิเคชันดอสที่ทำงานเกี่ยวกับการติดต่อทางเครือข่าย ภายใต้เอ็มเอสดอสวินโดว์ของไมโครซอฟต์วินโดว์ จำเป็นต้องป้องกันการเปลี่ยนการใช้งานระหว่างแอฟพลิเคชัน โดยการใช้อินเตอร์รัปต์ หมายเลข 2Fh ฟังก์ชัน 1601h (Begin Critical Section) และอินเตอร์รัปต์ หมายเลข 2Fh ฟังก์ชัน 1681h (End Critical Section) การใช้อินเตอร์รัปต์นี้ เพื่อป้องกันไม่ให้แอฟพลิเคชันอื่น เข้ามาใช้ทรัพยากรในระบบขณะที่แอฟพลิเคชันที่เรียกใช้อินเตอร์รัปต์นี้กำลังใช้งานอยู่

การใช้ไมโครซอฟต์ซี เวอร์ชัน 7.0 ในการพัฒนาโปรแกรม

ในการพัฒนาแอฟพลิเคชันบนดอส โดยใช้ไมโครซอฟต์ซี (Microsoft C) ในการพัฒนาโปรแกรม จะใช้โปรแกรม make ช่วยในการคอมไพล์โปรแกรม

โปรแกรม make จะทำการอ่าน makefile เพื่อทำการคอมไพล์ และลิงค์ โปรแกรมนอกจากนี้ โปรแกรม make ยังสามารถทำการตรวจสอบวันและเวลาที่สร้างหรือแก้ไขแฟ้มโปรแกรมต้นฉบับนั้น ๆ เพื่อทำการคอมไพล์ใหม่เฉพาะแฟ้มโปรแกรมต้นฉบับที่มีการแก้ไขเท่านั้น ซึ่งทำให้พัฒนาโปรแกรมได้เร็วขึ้น โดยเฉพาะโปรแกรมที่มี ออบเจ็ค ไฟล์ (object file) จำนวนมาก

สำหรับการใช้ไมโครซอฟต์ซี เวอร์ชัน 7.0 ในการพัฒนาโปรแกรม ต้องกำหนดสิ่งแวดล้อม (environment) ของระบบเพื่อสามารถอ้างถึง คำสั่งคอมไพล์และลิงค์ นอกจากนี้สามารถอ้างถึง อินคลูด ไฟล์ (include file) และ ไลเบรารี ไฟล์ (library file) ที่จำเป็นทั้งในส่วนของไมโครซอฟต์ซีเอง และของแอฟพลิเคชัน