

บทที่ 3

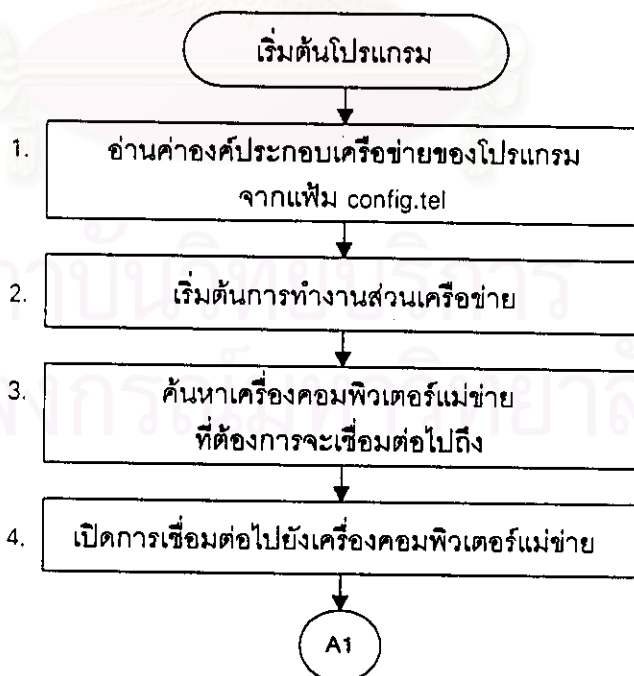
ส่วนประกอบการทำงานและการพัฒนาโปรแกรม

ในบทนี้กล่าวถึงส่วนประกอบการทำงานและการพัฒนาปรับปรุงโปรแกรมเลียนแบบเทอร์มินัลภาษาไทย CUTE ต้นฉบับเวอร์ชัน 2.03 (CUTE 2.03) จนได้โปรแกรมเลียนแบบเทอร์มินัลภาษาไทย CUTE เวอร์ชัน 2.10 (CUTE 2.10) ในส่วนเชื่อมต่อเครือข่าย และให้สามารถใช้งานภายใต้เอ็มเอสดีเอสวินโดว์ของไมโครซอฟต์วินโดว์ เวอร์ชัน 3.11

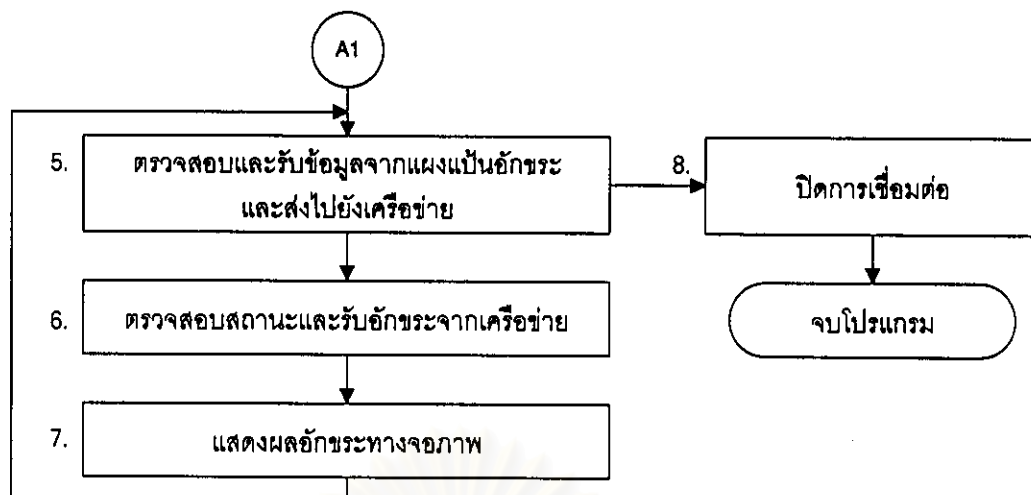
การพัฒนาปรับปรุงส่วนเชื่อมต่อเครือข่ายสำหรับโปรแกรม CUTE 2.10

จากโปรแกรมเลียนแบบเทอร์มินัลภาษาไทย CUTE 2.03 ซึ่งเป็นโปรแกรมต้นแบบในการพัฒนานี้ ได้พบปัญหาบางประการเกี่ยวกับการเชื่อมต่อกับเครื่องแม่ข่ายต่าง ๆ ดังนั้นจึงได้ทำการปรับปรุงส่วนเชื่อมต่อเครือข่าย โดยการพัฒนาส่วนของการเจรจาตัวเลือกของโปรแกรมเลียนแบบเทอร์มินัลภาษาไทย CUTE 2.03 ให้ดีขึ้น

ขั้นตอนการทำงานของโปรแกรม CUTE ในส่วนเชื่อมต่อกับโพรโตคอล ทีซีพี/ไอพี สามารถแสดงได้ดังรูปที่ 3.1 (เสกสรรค์,2538)



รูปที่ 3.1 แสดงขั้นตอนการทำงานของโปรแกรม CUTE ในส่วนเชื่อมต่อกับโพรโตคอล ทีซีพี/ไอพี



รูปที่ 3.1 แสดงขั้นตอนการทำงานของโปรแกรม CUTE ในส่วนเชื่อมต่อกับ
โพรโตคอล ทีซีพี/ไอพี (ต่อ)

จากรูปที่ 3.1 การวิจัยนี้จะทำการเพิ่มส่วนของการเจรจาตัวเลือก ในขั้นตอนที่ 4 คือ ขั้นตอนเปิดการเชื่อมต่อไปยังเครื่องแม่ข่าย และขั้นตอนที่ 6 คือ ขั้นตอนการตรวจสอบสถานะและรับอักขระจากเครื่องข่าย โดยเพิ่มฟังก์ชันในแต่ละขั้นตอนดังนี้

1. ส่วนเปิดการเชื่อมต่อกับเครื่องคอมพิวเตอร์แม่ข่าย

ทำการส่งข้อมูลเพื่อทำการขอการติดต่อกับเครื่องแม่ข่าย โดยส่งคำสั่งการเจรจาตัวเลือกไปยังเครื่องแม่ข่ายที่ต้องการติดต่อ ฟังก์ชันที่เพิ่มในขั้นตอนนี้คือ

ฟังก์ชัน start_negotiate

ฟังก์ชันนี้จะเป็นการขอเริ่มการติดต่อกับเครื่องแม่ข่าย โดยทำการส่งคำสั่งการเจรจาตัวเลือกไปยังเครื่องแม่ข่าย การเรียกใช้ฟังก์ชันมีรูปแบบดังนี้

```
#include variable.h
start_negotiate(tw)
struct tinfo *tw; /* Pointer to telnet information record */
```

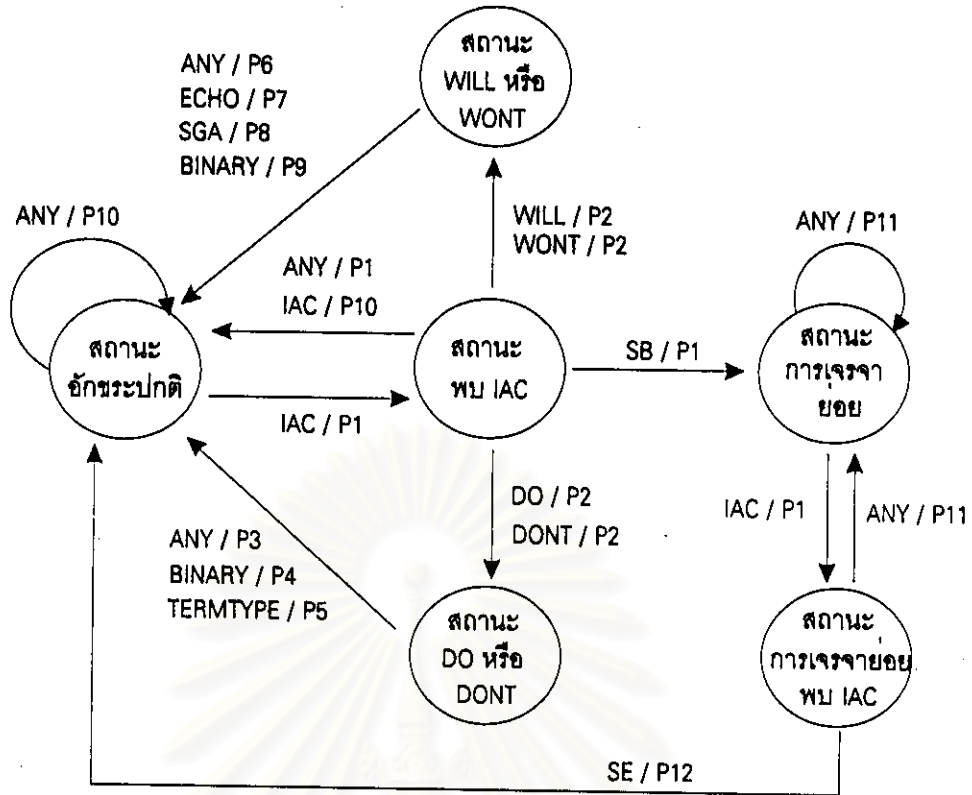
tw หมายถึงตัวชี้ไปยังโครงสร้างของ tinfo

ซึ่งก่อนการเรียกใช้ฟังก์ชันนี้ จะต้องทำการกำหนดค่าเริ่มต้นให้กับตัวแปร tw โดยเรียกใช้ ฟังก์ชัน Init_tw ซึ่งโครงสร้างข้อมูลของ tinfo ประกอบด้วย

```
struct tinfo {
    int pnum,          /* port number associated */
    width,            /* width of the window */
    rows,             /* Number of rows in the window */
    telstate,         /* telnet state for this connection */
    substate,         /* telnet subnegotiation state */
    unsigned int mapoutput:1, /* are we mapping the characters output for this window */
    termsent:1,       /* has terminal type been sent? */
    ibinary:1,        /* negotiate for binary traffic out */
    iwantbinary:1,    /* flag to indicate whether we asked for binary transmission out */
    ubinary:1,        /* negotiate for binary traffic in */
    uwantbinary:1,    /* flag to indicate whether we asked for binary transmission in */
    igoahead:1,       /* negotiate for suppress go-ahead */
    ugoahead:1,       /* negotiate for his suppress go-ahead */
    echo:1,           /* line mode or echo mode? */
};
```

2. ส่วนตรวจสอบสถานะและรับอักขระจากเครือข่าย

ส่วนตรวจสอบสถานะและรับอักขระจากเครือข่ายนี้ ทำการตรวจสอบข้อมูลที่ได้รับมาจากเครื่องแม่ข่ายที่เข้ามาทีละไบต์ โดยอาศัยแนวทางการกำหนดสถานะและฟังก์ชันที่ต้องทำในแต่ละสถานะให้เหมาะสม ซึ่งขั้นตอนนี้เป็นขั้นตอนช่วงของการทำการเจรจาตัวเลือก ตามเครื่องสถานะจำกัด (finite state machine) ดังรูปที่ 3.2



รูปที่ 3.2 แสดงเครื่องสถานะจำกัด สำหรับโปรแกรม CUTE 2.10

รูปที่ 3.2 แสดงเครื่องสถานะจำกัด สำหรับโปรแกรม CUTE 2.10 ซึ่งประกอบด้วยขั้นตอนการเจรจาตัวเลือก รวมไปถึงสถานะ ข้อมูล กระบวนการ และทิศทางการเปลี่ยนแปลงสถานะ ซึ่งมีกระบวนการต่าง ๆ ที่สำคัญดังแสดงในตารางที่ 3.1

กระบวนการทำงาน	คำอธิบาย
P1	เปลี่ยนสถานะเท่านั้น
P2	เปลี่ยนสถานะ และเก็บรหัสขอการรับบริการ เช่น DO DONT WILL หรือ WONT ขึ้นอยู่กับข้อมูลที่ได้รับ
P3	ไม่ต้องตอบสนองเครื่องแม่ข่าย กรณีที่ ร้องขอ หรือ ปฏิเสธ การทำใด ๆ
P4	ตอบสนองเครื่องแม่ข่ายโดยกรณีที่ ร้องขอ หรือ ปฏิเสธ การทำ BINARY
P5	ตอบสนองเครื่องแม่ข่ายโดยกรณีที่ ร้องขอ หรือ ปฏิเสธ การทำ TERMTYPE
P6	ไม่ต้องตอบสนองเครื่องแม่ข่าย กรณีที่ ยินยอม หรือ ไม่ยินยอม การทำใด ๆ

ตารางที่ 3.1 แสดงกระบวนการทำงาน ในเครื่องสถานะจำกัด สำหรับโปรแกรม CUTE 2.10

กระบวนการทำงาน	คำอธิบาย
P7	ตอบสนองเครื่องแม่ข่ายโดยกรณีที่ยินยอม หรือไม่ยินยอม การทำ ECHO
P8	ตอบสนองเครื่องแม่ข่ายโดยกรณีที่ยินยอม หรือไม่ยินยอม การทำ SGA
P9	ตอบสนองเครื่องแม่ข่ายโดยกรณีที่ยินยอม หรือไม่ยินยอม การทำ BINARY
P10	แสดงอักขระที่ได้รับมาออกทางจอภาพ
P11	เปลี่ยนสถานะ และเก็บข้อมูลที่ได้รับ เพื่อทำการเจรจาตัวเลือกย่อย (subnegotiate)
P12	ทำการวิเคราะห์ข้อมูลที่เข้ามา ในขั้นตอนการเจรจาตัวเลือกย่อย เช่น ส่งชนิดของเทอร์มินัล ตัวอย่างเช่น vt100 หรือ vt200 ไปยังเครื่องแม่ข่าย

ตารางที่ 3.1 แสดงกระบวนการทำงาน ในเครื่องสถานะจำกัด สำหรับโปรแกรม CUTE 2.10 (ต่อ)

ตัวอย่างเช่น เมื่อโปรแกรมเลียนแบบเทอร์มินัลภาษาไทย CUTE 2.10 ได้รับข้อมูลจากเครื่องข่าย ซึ่งจะอยู่ที่สถานะอักขระปกติ จะวิเคราะห์แต่ละไบต์ ถ้าเป็นอักขระปกติ จะอ่านไบต์ถัด ๆ ไปจนกว่าจะหมดข้อมูลชุดนั้น แต่ถ้าพบว่า มี อักขระ IAC จะทำการย้ายสถานะไปที่ สถานะพบ IAC โดยทำกระบวนการ P1 จากนั้นทำการวิเคราะห์ไบต์ที่ 2 ถ้าไบต์ที่ 2 พบว่าเป็น WILL จะทำการเปลี่ยนสถานะเป็น สถานะ WILL หรือ WONT จากนั้นทำการวิเคราะห์ไบต์ที่ 3 ถ้าไบต์ที่ 3 เป็น ECHO ซึ่งสามารถวิเคราะห์ได้ว่าข้อมูลที่ส่งมาจาก เครื่องแม่ข่ายคือ IAC WILL ECHO หมายความว่าเครื่องแม่ข่ายยินยอมให้ทำการ echo โปรแกรม CUTE 2.10 จะทำกระบวนการ P7 ซึ่งจะทำการส่งข้อมูล IAC DO ECHO ไปยังเครื่องแม่ข่าย เพื่อบอกว่าจะทำการ echo เป็นต้น

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ซึ่งขั้นตอนของการทำการเจรจาตัวเลือกนี้ ได้มีการปรับปรุงเพิ่มฟังก์ชันที่สำคัญดังนี้

ฟังก์ชัน negotiate

ฟังก์ชันนี้ตรวจสอบข้อมูลที่ได้รับมาจากเครื่องแม่ข่ายแต่ละไบต์ โดยตรวจสอบว่าข้อมูลที่เข้ามาเป็นข้อมูลอักขระปกติหรือข้อมูลที่เป็นคำสั่งการเจรจาตัวเลือก ถ้าเป็นข้อมูลที่เป็นอักขระปกติก็ส่งไปแสดงผลที่หน้าจอ แต่ถ้าเป็นคำสั่งการเจรจาตัวเลือก จะตรวจสอบตามเครื่องสถานะจำกัด ดังในรูปที่ 3.2 แสดงเครื่องสถานะจำกัด สำหรับโปรแกรม CUTE 2.10 ซึ่งการเรียกใช้ฟังก์ชัน มีรูปแบบดังนี้

```
#include variable.h
negotiate(tw,st,cnt)
struct tinfo *tw; /* Pointer to telnet information record */
unsigned char *st; /* Pointer to input buffer */
int cnt; /* number of byte read */
```

*tw หมายถึงตัวชี้ไปยังโครงสร้างของ tinfo
 *st หมายถึงตัวชี้ที่ชี้ไปยังบัฟเฟอร์ที่จะเก็บข้อมูลจากการอ่านชั่วคราว
 cnt หมายถึงจำนวนไบต์ที่อ่านได้จากเครือข่าย

สถาบันวิทยบริการ
 จุฬาลงกรณ์มหาวิทยาลัย

ขั้นตอนการติดต่อระหว่าง โปรแกรม CUTE 2.10 กับเครื่องแม่ข่าย

ขั้นตอนการติดต่อระหว่างโปรแกรม CUTE 2.10 กับเครื่องแม่ข่าย จนกระทั่งได้เครื่องหมาย "login : " มี 2 ขั้นตอนหลัก ดังนี้

1. การเริ่มติดต่อ

ขั้นตอนนี้ฝั่งไคล์เอ็นต์ คือเครื่องไมโครคอมพิวเตอร์ที่ใช้โปรแกรม CUTE 2.10 ทำการขอติดต่อกับเครื่องแม่ข่าย โดยส่งคำสั่งการเจรจาตัวเลือกกับเทลเน็ตเซิร์ฟเวอร์ ให้ทำการ ECHO และทำ SGA พร้อมทั้งแจ้งให้เทลเน็ตเซิร์ฟเวอร์ทราบ ว่า ไคล์เอ็นต์ยินดีให้ข้อมูลเกี่ยวกับรายละเอียดขนาดของจอภาพแสดงผล

2. การเจรจาตัวเลือก

ขั้นตอนนี้ทำหลังจากที่ไคล์เอ็นต์ได้ส่งคำสั่งการเจรจาตัวเลือก โดยเทลเน็ตเซิร์ฟเวอร์จะส่งคำสั่งการเจรจาตัวเลือกตอบกลับมายังฝั่งไคล์เอ็นต์ จากนั้นจะมีการโต้ตอบกันระหว่างไคล์เอ็นต์กับเทลเน็ตเซิร์ฟเวอร์ ซึ่งขั้นตอนการติดต่อระหว่างไคล์เอ็นต์กับเทลเน็ตเซิร์ฟเวอร์จะขึ้นอยู่กับชนิดของเครื่องแม่ข่าย ซึ่งในที่นี้จะขอยกตัวอย่างขั้นตอนการติดต่อกันระหว่างโปรแกรม CUTE 2.10 กับเครื่องแม่ข่ายที่ชื่อ unixgrp ซึ่งเป็นเครื่องเวอร์กสเตชัน ของบริษัท ซันไมโครซิสเต็มส์ รุ่น SPARCStation 10 ซึ่งใช้ระบบปฏิบัติการ Solaris เวอร์ชัน 2.4 โดยมีขั้นตอนการติดต่อดังนี้

2.1 ฝั่งเซิร์ฟเวอร์ขอทราบชนิดของเทอร์มินัล (Term Type) จากฝั่งไคล์เอ็นต์ และทางฝั่งไคล์เอ็นต์ตอบกลับว่ายินดีให้ข้อมูลของชนิดของเทอร์มินัล

2.2 ฝั่งเซิร์ฟเวอร์แจ้งให้ฝั่งไคล์เอ็นต์ทราบว่า ยินดีทำ ECHO ให้ และฝั่งไคล์เอ็นต์ตอบกลับว่าต้องการทำ ECHO

2.3 ฝั่งเซิร์ฟเวอร์แจ้งให้ฝั่งไคล์เอ็นต์ทราบว่า ยินดีทำ SGA ให้ และฝั่งไคล์เอ็นต์ตอบกลับว่าต้องการทำ SGA

2.4 ฝั่งเซิร์ฟเวอร์แจ้งให้ฝั่งไคลเอ็นต์ทราบว่า ไม่ต้องส่งข้อมูลเกี่ยวกับรายละเอียดขนาดของจอภาพแสดงผล

2.5 ฝั่งเซิร์ฟเวอร์ต้องการชนิดเทอร์มินัล และฝั่งไคลเอ็นต์ ส่งชนิดเทอร์มินัล เป็น VT100 ไปให้

2.6 ฝั่งเซิร์ฟเวอร์ส่งข้อความ ซึ่งข้อความนี้ขึ้นอยู่กับโปรแกรมระบบจัดการบนฝั่งเครื่องแม่ข่าย เพื่อแสดงว่าไคลเอ็นต์ติดต่อกับเครื่องแม่ข่ายตัวใด อาทิเช่น Unix(r) System V Release 4.0 (unixgrp) หรือ Digital Unix (Columbia) เป็นต้น

2.7 ฝั่งเซิร์ฟเวอร์แจ้งให้ฝั่งไคลเอ็นต์ทราบว่า ต้องการทำ ECHO จากนั้นส่งเครื่องหมาย "login : " ไปยังฝั่งไคลเอ็นต์

ซึ่งจากขั้นตอนการรับส่งข้อมูล เพื่อการติดต่อระหว่างโปรแกรม CUTE 2.10 กับ เครื่องแม่ข่ายดังกล่าวข้างต้น สามารถแสดงได้ดังรูปที่ 3.3 และรูปที่ 3.4

โปรแกรม CUTE 2.10

(PC / Client)



Telnet Server

(Host / Server)



ขั้นตอนที่	ข้อมูลและทิศทางการเจรจาตัวเลือก ระหว่าง โปรแกรม CUTE 2.10 กับ เทลเน็ตเซิร์ฟเวอร์	คำอธิบาย
1	-----> IAC DO ECHO IAC DO SGA IAC WILL NAWS	โปรแกรม CUTE 2.10 เริ่ม การติดต่อ โดยขอให้ เซิร์ฟเวอร์ ทำ ECHO และ SGA และจะให้รายละเอียด ขนาดของจอภาพแสดงผล

รูปที่ 3.3 แสดงขั้นตอนการเริ่มติดต่อ

โปรแกรม CUTE 2.10
(PC / Client)



Telnet Server
(Host / Server)



ขั้นตอนที่	ข้อมูลและทิศทางการเจรจาตัวเลือก ระหว่าง โปรแกรม CUTE 2.10 กับเทลเน็ตเซิร์ฟเวอร์	คำอธิบาย
2.1	<pre> <----- IAC DO TERMTYPE -----> IAC WILL TERMTYPE </pre>	เซิร์ฟเวอร์ขอทราบชนิดของเทอร์มินัล และไคล์เอ็นต์ ยินดีให้ข้อมูลของเทอร์มินัล
2.2	<pre> <----- IAC WILL ECHO -----> IAC DO ECHO </pre>	เซิร์ฟเวอร์แจ้งว่ายินดีทำ ECHO ให้ และไคล์เอ็นต์ตอบกลับว่าต้องการทำ ECHO
2.3	<pre> <----- IAC WILL SGA -----> IAC DO SGA </pre>	เซิร์ฟเวอร์แจ้งว่ายินดีทำ SGA ให้ และไคล์เอ็นต์ตอบกลับว่าต้องการทำ SGA
2.4	<pre> <----- IAC DONT NAWS </pre>	เซิร์ฟเวอร์แจ้งว่า ไม่ต้องส่งรายละเอียดขนาดของจอภาพแสดงผล
2.5	<pre> <----- IAC SB TERMTYPE ECHO IAC SE -----> IAC SB TERMTYPE VT100 IAC SE </pre>	เซิร์ฟเวอร์ต้องการชนิดของเทอร์มินัล และไคล์เอ็นต์ส่งชนิดของเทอร์มินัลเป็น VT100 ไปให้

รูปที่ 3.4 แสดงขั้นตอนการเจรจาตัวเลือก

ขั้นตอนที่	ข้อมูลและทิศทางการเจรจาตัวเลือก ระหว่าง โปรแกรม CUTE 2.10 กับเทลเน็ตเซิร์ฟเวอร์	คำอธิบาย
2.6	<pre> <----- Unix(r) System V Release 4.0 (unixgrp) </pre>	เซิร์ฟเวอร์ทำการส่งข้อความเพื่อแสดงให้เห็นว่าไคลเอ็นต์ติดต่อกับเครื่องแม่ข่ายตัวใด
2.7	<pre> <----- IAC DO ECHO <----- login : </pre>	เซิร์ฟเวอร์ต้องการทำ ECHO จากนั้นจะส่งเครื่องหมาย "login : " ไปยังฝั่งไคลเอ็นต์

รูปที่ 3.4 แสดงขั้นตอนการเจรจาตัวเลือก (ต่อ)

การพัฒนาปรับปรุงโปรแกรม CUTE 2.10 ให้สามารถใช้งานภายใต้เอ็มเอสดอสวินโดว์ของไมโครซอฟต์วินโดว์ เวอร์ชัน 3.11

ในการพัฒนาปรับปรุงโปรแกรม CUTE 2.10 เพื่อให้ทำงานได้ภายใต้เอ็มเอสดอสวินโดว์ของไมโครซอฟต์วินโดว์ เวอร์ชัน 3.11 ได้ทำการเพิ่มเติมบางฟังก์ชันในขั้นตอนที่ 2 จากรูปที่ 3.1 แสดงขั้นตอนการทำงานของโปรแกรม CUTE ในส่วนเชื่อมต่อกับโพรโตคอล ทีซีพี/ไอพี คือ ขั้นตอนเริ่มต้นการทำงานส่วนเครือข่าย โดยเพิ่มเติมฟังก์ชันดังนี้

1. ฟังก์ชัน check_win

เป็นฟังก์ชันสำหรับตรวจสอบว่าโปรแกรม CUTE 2.10 ทำงานอยู่ ภายใต้เอ็มเอสดอสวินโดว์ของไมโครซอฟต์วินโดว์หรือไม่ โดยใช้อินเตอร์รัปต์ หมายเลข 2Fh ฟังก์ชัน 1600h โดยผลที่ได้จากการตรวจสอบบรรจุไว้ในรีจิสเตอร์ AL ซึ่งสามารถตรวจสอบผลได้ดังนี้

รีจิสเตอร์ AL มีค่าเป็น 00h หมายความว่า โปรแกรมไม่ได้ทำงานอยู่ภายใต้ไมโครซอฟต์วินโดว์

รีจิสเตอร์ AL มีค่าเป็น 01h หรือ FFh หมายความว่าทำงานอยู่ภายใต้ไมโครซอฟต์วินโดว์ เวอร์ชัน 2.x

รีจิสเตอร์ AL มีค่าเป็น 80h หมายความว่าทำงานอยู่ภายใต้ไมโครซอฟต์แวร์ วินโดว์ เวอร์ชัน 3.x ที่ไม่ใช่ Enhance mode

รีจิสเตอร์ AL มีค่าที่ไม่ใช่ 00h, 01h, 80h หรือ FF แสดงว่าทำงานอยู่ภายใต้ไมโครซอฟต์แวร์วินโดว์ เวอร์ชัน 3.x ใน Enhance mode ซึ่งการเรียกใช้ฟังก์ชันมีรูปแบบดังนี้

```
#include <dos.h>

win_flag = check_win( );
```

win_flag หมายถึงตัวแปรบ่งชี้ว่าโปรแกรมทำงานอยู่ที่เอ็มเอสดอสวินโดว์หรือไม่

2. ฟังก์ชัน begin_critical

เป็นฟังก์ชันที่ป้องกันไม่ให้แอฟพลิเคชันอื่นเข้ามาแย่งใช้ทรัพยากรของระบบ และป้องกันไม่ให้เกิดการเปลี่ยนงาน (task switching) ขณะที่โปรแกรม CUTE 2.10 กำลังทำงานอยู่ภายใต้เอ็มเอสดอสวินโดว์ของไมโครซอฟต์แวร์วินโดว์ เวอร์ชัน 3.x ใน Enhance mode ซึ่งฟังก์ชันนี้ เรียกใช้อินเตอร์รัปต์ 2Fh ฟังก์ชัน 1601h การเรียกใช้ฟังก์ชันมีรูปแบบดังนี้

```
#include <dos.h>

begin_critical( );
```

3. ฟังก์ชัน end_critical

ฟังก์ชันนี้จะใช้คู่กับฟังก์ชัน begin_critical() เมื่อต้องการเลิกใช้งานโปรแกรม CUTE 2.10 ที่ทำงานอยู่ภายใต้เอ็มเอสดอสวินโดว์ ของไมโครซอฟต์แวร์วินโดว์ เวอร์ชัน 3.x ใน Enhance mode ซึ่งฟังก์ชันนี้ จะเรียกใช้อินเตอร์รัปต์ 2Fh ฟังก์ชัน 1681h การเรียกใช้ฟังก์ชันมีรูปแบบดังนี้

```
#include <dos.h>

end_critical( );
```

ในการพัฒนาปรับปรุงโปรแกรมเลียนแบบเทอร์มินัลภาษาไทย CUTE เพื่อให้มีประสิทธิภาพในส่วนเชื่อมต่อกับเครื่องแม่ข่ายต่าง ๆ บนระบบเครือข่ายที่มี โพรโตคอล ทีซีพี/ไอพี และมีเทคโนโลยีไฟร์วอลล์ ได้สมบูรณ์ขึ้นนั้น โดยทำการเพิ่มส่วนของการเจรจาตัวเลือก และใช้หลักการของเครื่องสถานะจำกัด สำหรับการพัฒนาโปรแกรมเลียนแบบเทอร์มินัลภาษาไทย CUTE ให้สามารถทำงานภายใต้เอ็มเอสดีเอสวินโดว์ ของไมโครซอฟต์วินโดว์ เวอร์ชัน 3.11 โดยใช้ฟังก์ชันของอินเตอร์พรีตที่สามารถป้องกันการแย่งกันใช้ทรัพยากรต่าง ๆ ขณะที่ใช้งานแอปพลิเคชันคอสทำงานภายใต้เอ็มเอสดีเอสวินโดว์ ของไมโครซอฟต์วินโดว์ เวอร์ชัน 3.11 ซึ่งผลจากการทดสอบหลังจากทำการพัฒนาปรับปรุงโปรแกรมเลียนแบบเทอร์มินัลภาษาไทย CUTE จนได้โปรแกรม CUTE เวอร์ชัน 2.10 ได้กล่าวไว้ในบทที่ 4 เรื่องการทดสอบโปรแกรม



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย