



บริการเพิ่มเติมด้านการศึกษาในระบบอินเทอร์เน็ต
VALUE-ADDED EDUCATION SERVICE TO THE INTERNET



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมคอมพิวเตอร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าในรูปแบบใดก็ตาม และขอสงวนสิทธิ์ในข้อความนี้ไว้เพื่อใช้ในการอ้างอิงเท่านั้น

บริการเพิ่มเติมด้านการศึกษาในระบบอินเทอร์เน็ต
VALUE-ADDED EDUCATION SERVICE TO THE INTERNET



**ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมคอมพิวเตอร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2537**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บริการเพิ่มเติมด้านการศึกษาในระบบอินเตอร์เน็ต

กรรภัทร์ สุทธิดารา

ชัยทัต หวังสัมพันธ์

ศ.ดร.ศรัทธาภัค จามรมาน อาจารย์ที่ปรึกษา

บทคัดย่อ

ในวิทยานิพนธ์ฉบับนี้ เรียบเรียงขึ้นจากผลงานโปรแกรมที่สร้างขึ้นเพื่อใช้ในการงานบริการเพิ่มเติม ด้านการศึกษาในระบบอินเทอร์เนต (Internet) เช่น การเลื่อนเวลาเรียน, การนัดเวลาเรียนใหม่, การส่ง การบ้าน, การส่งรับจดหมายหรือประกาศอื่นๆ โดยพัฒนาภายใต้ระบบปฏิบัติการยูนิกซ์ (UNIX) ที่เชื่อมโยงกับเครือข่ายอินเทอร์เน็ตภายใต้การสื่อสารระหว่างผู้ใช้กับเทอร์มินัล (terminal) แบบวินโดว (window) ซึ่ง จะทำให้การใช้งานเป็นไปได้สะดวกกว่าการใช้งานแบบชุดคำสั่ง (command line) เช่นสามารถเลือกผู้รับจด หมายผ่านเมนูแทนที่ต้องกรอกชื่อผู้รับโดยผู้ใช้อย่างซ้ำๆ ในการส่งจดหมาย โดยการติดต่อสื่อสารผ่าน ซอฟต์แวร์ (software) การจัดการกับตู้รับจดหมาย (mailbox) และการส่งจดหมายที่สามารถโปรแกรมได้ ซึ่งเป็นเครื่องมือที่ช่วยทำให้การจัดการรวมถึงการปรับปรุง เปลี่ยนแปลงรูปแบบที่ใช้ในการสื่อสารระหว่างผู้ใช้ กับเทอร์มินัลที่ผู้ใช้นั้นทำงานอยู่ได้สะดวกขึ้น เพียงทำการแก้ไขรูปแบบในแฟ้มซึ่งกำหนดรูปแบบให้เป็นไป ตามที่ผู้ใช้นั้นต้องการเท่านั้น

VALUE-ADDED EDUCATION SERVICE TO THE INTERNET

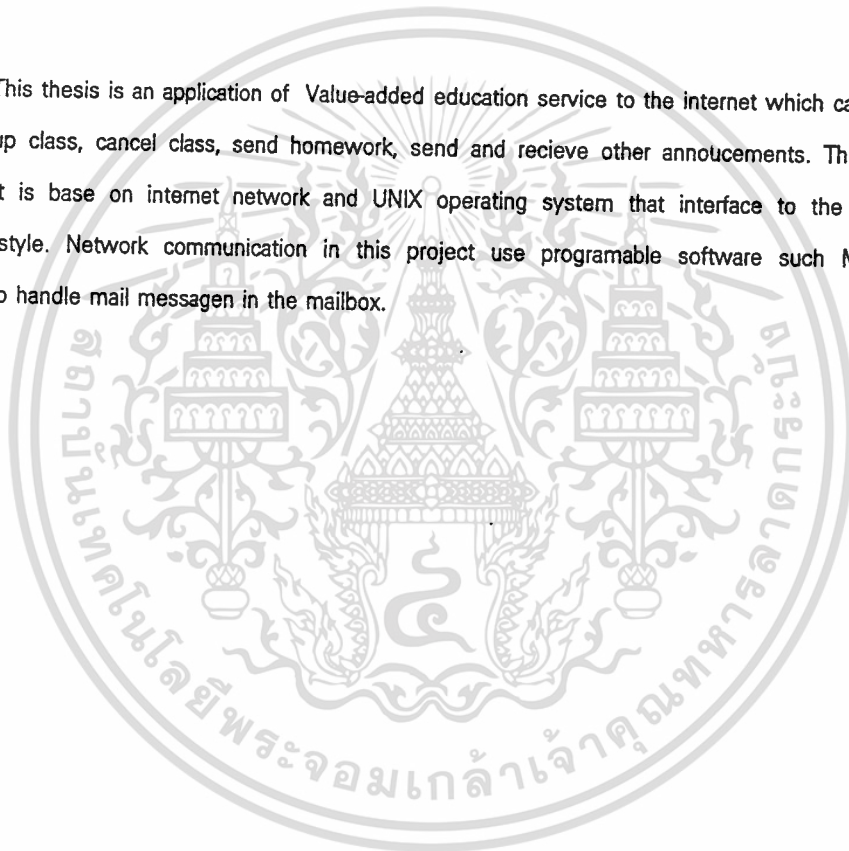
Korrapat Sutidara

Chaitad Huangsamphan

Professor Dr. Srisakdi Charmonman Advisor

Abstract

This thesis is an application of Value-added education service to the internet which can be use to make up class, cancel class, send homework, send and recieve other annoucements. This project enviroment is base on internet network and UNIX operating system that interface to the users in windows style. Network communication in this project use programable software such MH (Mail handling) to handle mail messagen in the mailbox.



ปริญญานิพนธ์ปีการศึกษา 2537

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง บริการเพิ่มเติมด้านการศึกษาในระบบอินเตอร์เน็ต

ผู้จัดทำ

นักศึกษาภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

1. นายกรภัทร์ สุทธิดารา รหัสประจำตัว 34101003

2. นายชัยทัต หวังสัมพันธ์ รหัสประจำตัว 34102085



.....อาจารย์ที่ปรึกษา

(.....)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎี	2
บทที่ 3 การสร้าง	34
บทที่ 4 การทดลอง และผลการทดลอง	133
บทที่ 5 บทวิจารณ์ และสรุป	138

กิตติกรรมประกาศ
หนังสืออ้างอิง



วัตถุประสงค์

จุดประสงค์ในโครงการนี้ เพื่อการเรียนรู้ระบบปฏิบัติการยูนิกซ์ และเครือข่ายอินเทอร์เน็ตที่เป็นเครือข่ายที่มีความสามารถในการสื่อสารได้ครอบคลุมพื้นที่ได้อย่างกว้างขวาง และทำการพัฒนาโปรแกรมขึ้นมาใช้กับระบบดังกล่าว โดยโปรแกรมมีจุดประสงค์ดังต่อไปนี้

1. เพื่อพัฒนาโปรแกรม สำหรับการติดต่อสื่อสารใช้สำหรับการศึกษา
 - 1.1 การเลื่อนการเรียนการสอน
 - 1.2 การนัดการเรียนการสอนใหม่
 - 1.3 การส่งการบ้าน
 - 1.4 การส่ง-รับ ประการต่างๆ
2. เพื่อการเรียนรู้ ขั้นตอนการทำงานต่างๆในระบบ
 - 2.1 การเขียนโปรแกรม ภาษาซีในระบบปฏิบัติการยูนิกซ์
 - 2.2 การทำงานในรูปแบบหน้าต่าง (window) โดยใช้เคอร์ส (curses) ร่วมกับภาษาซีในระบบปฏิบัติการยูนิกซ์
 - 2.3 การทำการสื่อสารระหว่างผู้ใช้กับเครื่องเทอร์มินัล (terminal)
3. เพื่อสร้างการเชื่อมต่อระหว่างผู้ใช้กับเครื่องเทอร์มินัล (terminal) ด้วยระบบเมนู

บทที่ 1

บทนำ

ในบทนี้จะกล่าวถึงวัตถุประสงค์ของโครงการเพื่อตั้งวัตถุประสงค์และดำเนินโครงการตามวัตถุประสงค์ที่ตั้งไว้

วัตถุประสงค์

จุดประสงค์ในโครงการนี้ เพื่อการเรียนรู้ระบบปฏิบัติการยูนิกซ์ และเครือข่ายอินเทอร์เน็ตที่เป็นเครือข่ายที่มีความสามารถในการสื่อสารได้ครอบคลุมพื้นที่ได้อย่างกว้างขวาง และทำการพัฒนาโปรแกรมขึ้นมาใช้กับระบบดังกล่าว โดยโปรแกรมมีจุดประสงค์ดังต่อไปนี้

1. เพื่อพัฒนาโปรแกรม สำหรับการติดต่อสื่อสารใช้สำหรับการศึกษา
 - 1.1 การเรียนการสอน
 - 1.2 การนัดการเรียนการสอนใหม่
 - 1.3 การส่งการบ้าน
 - 1.4 การส่งรับภาระต่างๆ
2. เพื่อการเรียนรู้ ขั้นตอนการทำงานต่างๆในระบบ
 - 2.1 การเขียนโปรแกรม ภาษาซีในระบบปฏิบัติการยูนิกซ์
 - 2.2 การทำงานในรูปแบบหน้าต่าง (window) โดยใช้เคอร์ส (curses) ร่วมกับภาษาซีในระบบปฏิบัติการยูนิกซ์
 - 2.3 การทำการสื่อสารระหว่างผู้ใช้กับเครื่องเทอร์มินัล (terminal)
3. เพื่อสร้างการเชื่อมต่อระหว่างผู้ใช้กับเครื่องเทอร์มินัล (terminal) ด้วยระบบเมนู

บทที่ 2

ทฤษฎีและหลักการ

ในบทนี้กล่าวถึงทฤษฎีของเครือข่ายอินเทอร์เน็ต(internet)และคำสั่งในการเขียนโปรแกรมสำหรับระบบปฏิบัติการยูนิกซ์(UNIX)รวมทั้งกล่าวถึงการตั้ง(set)ระบบด้วย 'stty'



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การต่อคอมพิวเตอร์เป็นเทอร์มินัล

การต่อเชื่อมคอมพิวเตอร์เข้าสู่อินเทอร์เน็ตโดยให้คอมพิวเตอร์เครื่องนั้นทำหน้าที่เป็นเพียงเทอร์มินัล จะมีรูปแบบการต่อเชื่อมที่ไม่สลับซับซ้อนและใช้อุปกรณ์จำนวนน้อยทั้งนี้เนื่องจากว่าต้องอาศัยศูนย์บริการเครือข่ายที่ต่อเชื่อมกับอินเทอร์เน็ตอยู่ก่อนแล้ว และผู้ใช้ก็จำเป็นต้องมีบัญชีประจำเครื่องที่ศูนย์บริการนั้น

ผู้ใช้เพียงแต่ขอเชื่อมโยงคอมพิวเตอร์ซึ่งอาจจะตั้งอยู่ที่บ้านหรือที่สำนักงานเข้ากับเครื่องที่ศูนย์บริการ โดยใช้สายโทรศัพท์ผ่านทางโมเด็ม และใช้ซอฟต์แวร์ ที่เรียกว่า “ตัวจำลองเทอร์มินัล” เพื่อทำหน้าที่แปลงคอมพิวเตอร์ให้เป็นเทอร์มินัล ตัวหนึ่งของคอมพิวเตอร์ ที่ศูนย์บริการ ซอฟต์แวร์จำลองเทอร์มินัล จะช่วยให้เราสามารถใช้อุปกรณ์ที่ศูนย์บริการเสมือนกับว่าได้นั่งทำงานอยู่ที่เครื่องนั้นโดยตรง รูปแบบของการต่อแบบนี้จะมีลักษณะดังรูปที่แสดง การต่อคอมพิวเตอร์ให้เป็นเทอร์มินัล

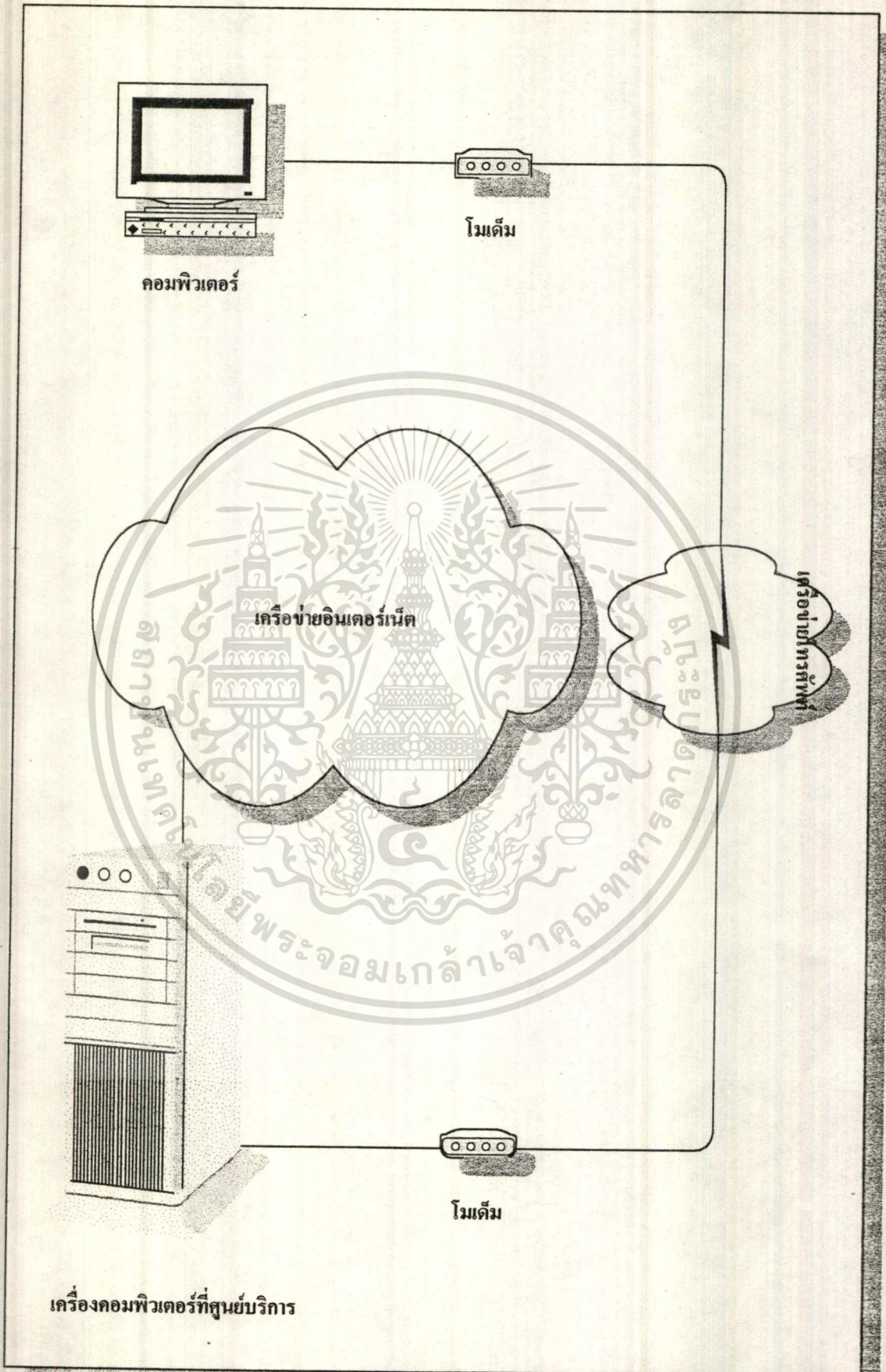
จากรูปจะแสดงให้เห็นว่าการใช้คอมพิวเตอร์เครื่องหนึ่ง ต่อเข้ากับโมเด็มที่พ่วงกับคู่สายโทรศัพท์ โดยที่โมเด็มจะต่ออยู่กับเครื่องคอมพิวเตอร์ผ่านทางพอร์ตนุกรม ทางด้านส่วนปลายของศูนย์บริการก็จะมีโมเด็มซึ่งจะเตรียมไว้แล้วเป็นตัวรับส่งข้อมูลกับโมเด็มของผู้ใช้ โมเด็มที่ศูนย์บริการจะเชื่อมต่อไปยังคอมพิวเตอร์ตัวบริการซึ่งโยงเข้าสู่อินเทอร์เน็ตอีกต่อหนึ่ง เพียงเท่านั้นผู้ใช้อีกจะได้ช่องทางเข้าสู่อินเทอร์เน็ตโดยสมบูรณ์

โดยสรุปแล้วผู้ใช้อาจเตรียมและเชื่อมต่ออุปกรณ์ต่างๆดังมีรายละเอียดของเครื่องคอมพิวเตอร์โมเด็ม ตลอดจนซอฟต์แวร์ดังรูป อุปกรณ์ที่ใช้เชื่อมต่อให้คอมพิวเตอร์เป็นเทอร์มินัล อันมีรายละเอียดดังต่อไปนี้

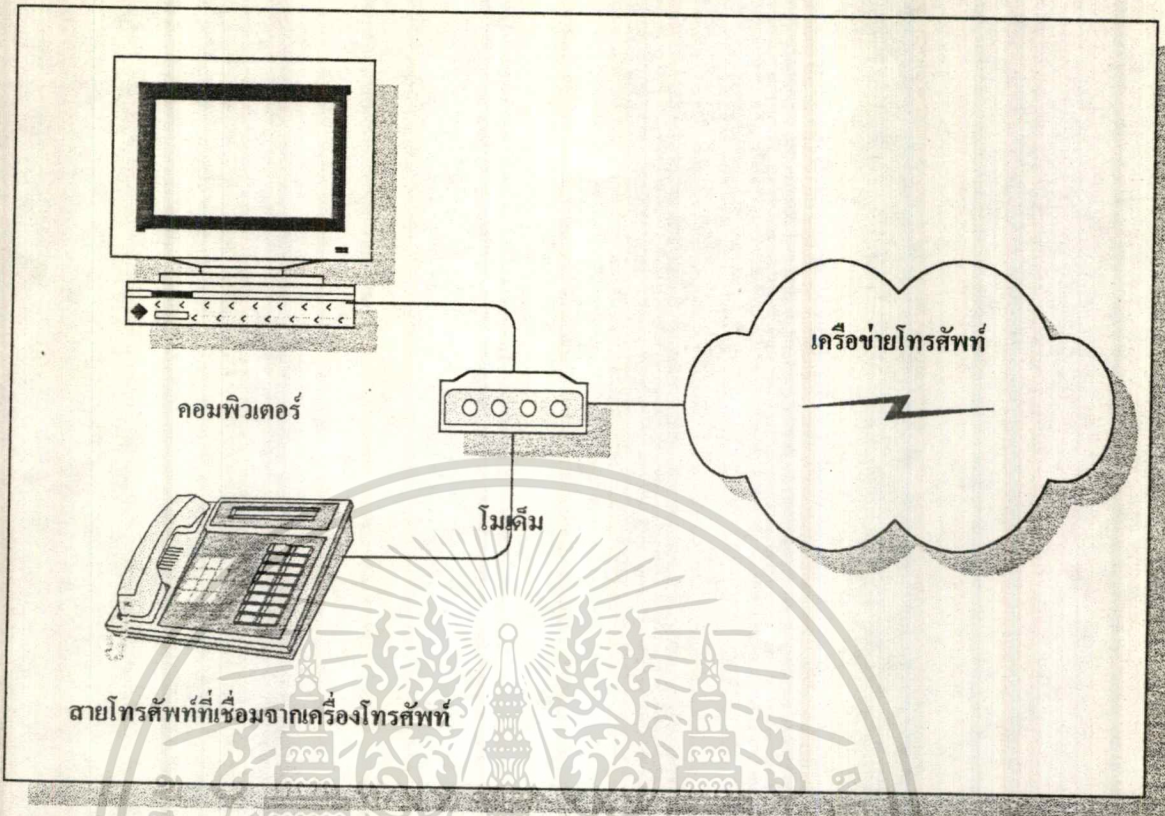
คอมพิวเตอร์ที่ใช้เป็นเทอร์มินัล เครื่องที่ใช้งานไม่จำกัดว่าจะต้องเป็นเครื่องยี่ห้อใด ผู้ใช้จึงสามารถเลือกใช้เครื่องได้ทั้ง เครื่องแบบ พีซี อาตาริ อาไมก้า หรือ แมคอินทอช เพียงแต่ต้องมีพอร์ตสื่อสารอนุกรมเพื่อต่อกับโมเด็มชนิดของคอมพิวเตอร์ที่เลือกใช้ส่งผลในด้านการหาซอฟต์แวร์มาทำงานเท่านั้น สำหรับในที่นี้จะขอยกตัวอย่างในกรณีที่ใช้พีซี เนื่องจากเป็นเครื่องที่ใช้กันอย่างแพร่หลาย

ซอฟต์แวร์จำลองเทอร์มินัล ซอฟต์แวร์ที่จำเป็นต้องใช้ คือซอฟต์แวร์ที่ทำหน้าที่จำลองคอมพิวเตอร์ให้เป็นเทอร์มินัลของคอมพิวเตอร์ที่ศูนย์บริการ ซอฟต์แวร์ประเภทนี้มีอยู่มากมาย และมีขีดความสามารถแตกต่างกันไป ตัวอย่างเช่น โปรคอม พลัส, เทเล็กซ์, พีซี เอ็นนิแวร์, ครอส ทอล์ค หรือ เคอร์มิต เป็นต้น

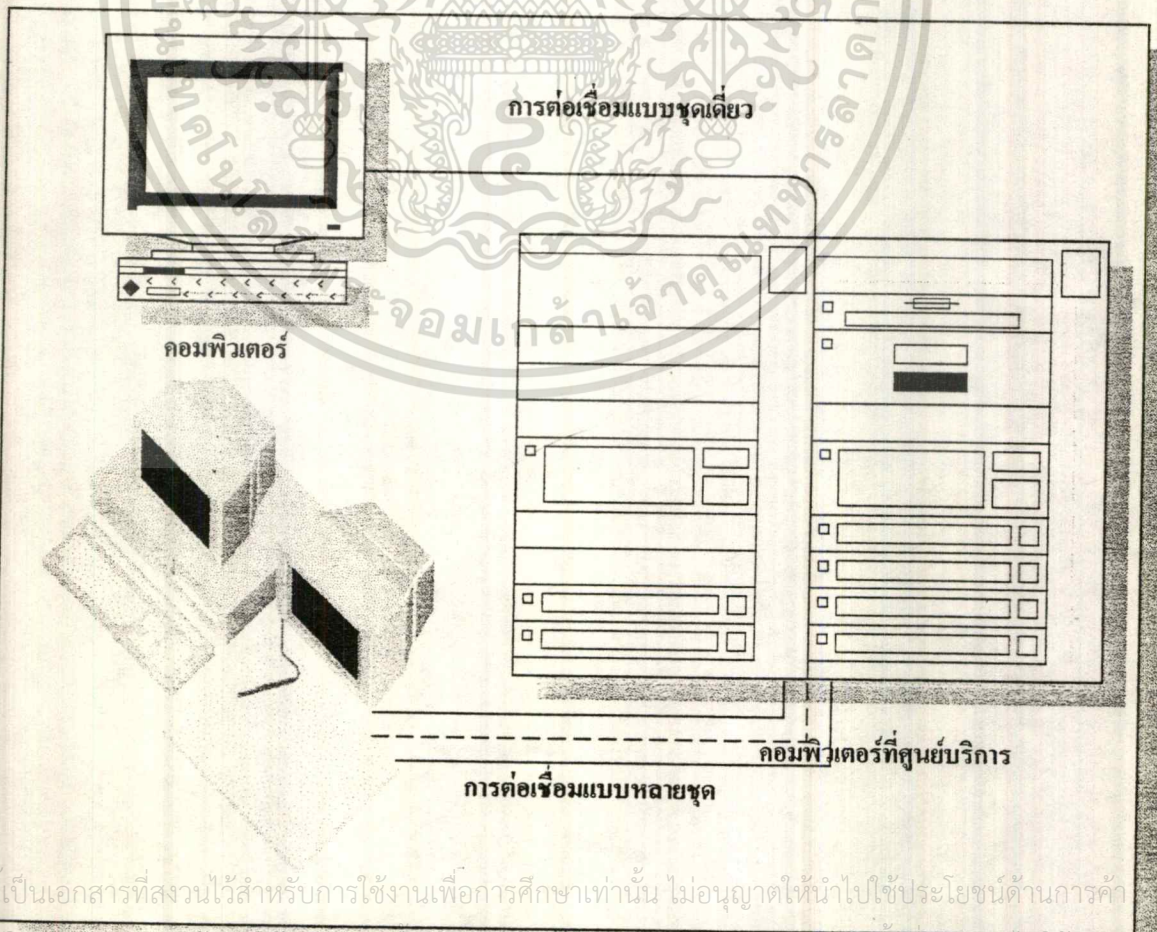
โมเด็ม ชนิดและความเร็วของโมเด็มมีความแตกต่างกันให้ท่านเลือกได้อย่างมากมาย และราคาก็แตกต่างกันไปด้วย โมเด็มที่มีจำหน่ายในท้องตลาดอาจมีทั้งที่ความเร็วถึง 28.8 กิโลบิตต่อวินาที และความเร็วลดต่ำลงไปเป็นลำดับเช่น 19.2, 14.4, 9.6, 4.8, 2.4 กิโลบิตต่อวินาที จนกระทั่งถึง 1.2 กิโลบิตต่อวินาทีในปัจจุบัน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
การเชื่อมต่อคอมพิวเตอร์ให้เป็นเทอร์มินัล
 ไม่ว่าการนี้ใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



อุปกรณ์ที่ใช้เชื่อมต่อให้กับเทอร์มินัล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้าม

เปรียบเทียบการต่อเชื่อมแบบชุดเดียวและหลายชุด

การเชื่อมต่อด้วยโปรโตคอล

การเชื่อมต่อแบบแรกที่กล่าวไปแล้วเหมาะสมกับผู้ใช้งานที่ต้องการเพียงการเชื่อมต่อแบบขั้นต้นซึ่งติดตั้งและดูแลระบบได้ง่าย และโปรแกรมที่ใช้งานก็ไม่สลับซับซ้อน หากแต่เป็นวิธีการที่มีข้อจำกัดคือเปิดใช้ระบบได้เพียงชุดเดียวเท่านั้น

ช่องทางในการใช้โฮสต์หรือพอเปรียบเทียบได้ว่าหนึ่งช่องทางก็คือหนึ่งจอภาพที่เชื่อมต่อกับโฮสต์ระบบที่มีช่องทางเพียงชุดเดียวก็จะมีเพียงจอภาพชุดเดียวในขณะใดขณะหนึ่ง แต่ระบบที่เป็นช่องทางหลายชุดจะสามารถเปิดได้หลายจอภาพเพื่อเชื่อมต่อกับโฮสต์ต่างๆ ได้หลายตัวพร้อมกัน หรือเชื่อมต่อกับโฮสต์ตัวเดียวกันได้หลายครั้ง นั่นหมายความว่าเราสามารถเปิดจอภาพเพื่อต่อกับโฮสต์ได้มากกว่าหนึ่งจอ ในขณะเวลาใดเวลาหนึ่งจะมีการแสดงจอเพียงชุดใดชุดหนึ่ง แต่ซอฟต์แวร์ที่ใช้งานในระบบนี้จะบีบฟังก์ชันให้เราสลับจอไปมาเพื่อเลือกจอที่ต้องการได้ ดังรูปที่แสดงเกี่ยวกับ การเปรียบเทียบการต่อเชื่อมแบบชุดเดียว และหลายชุด

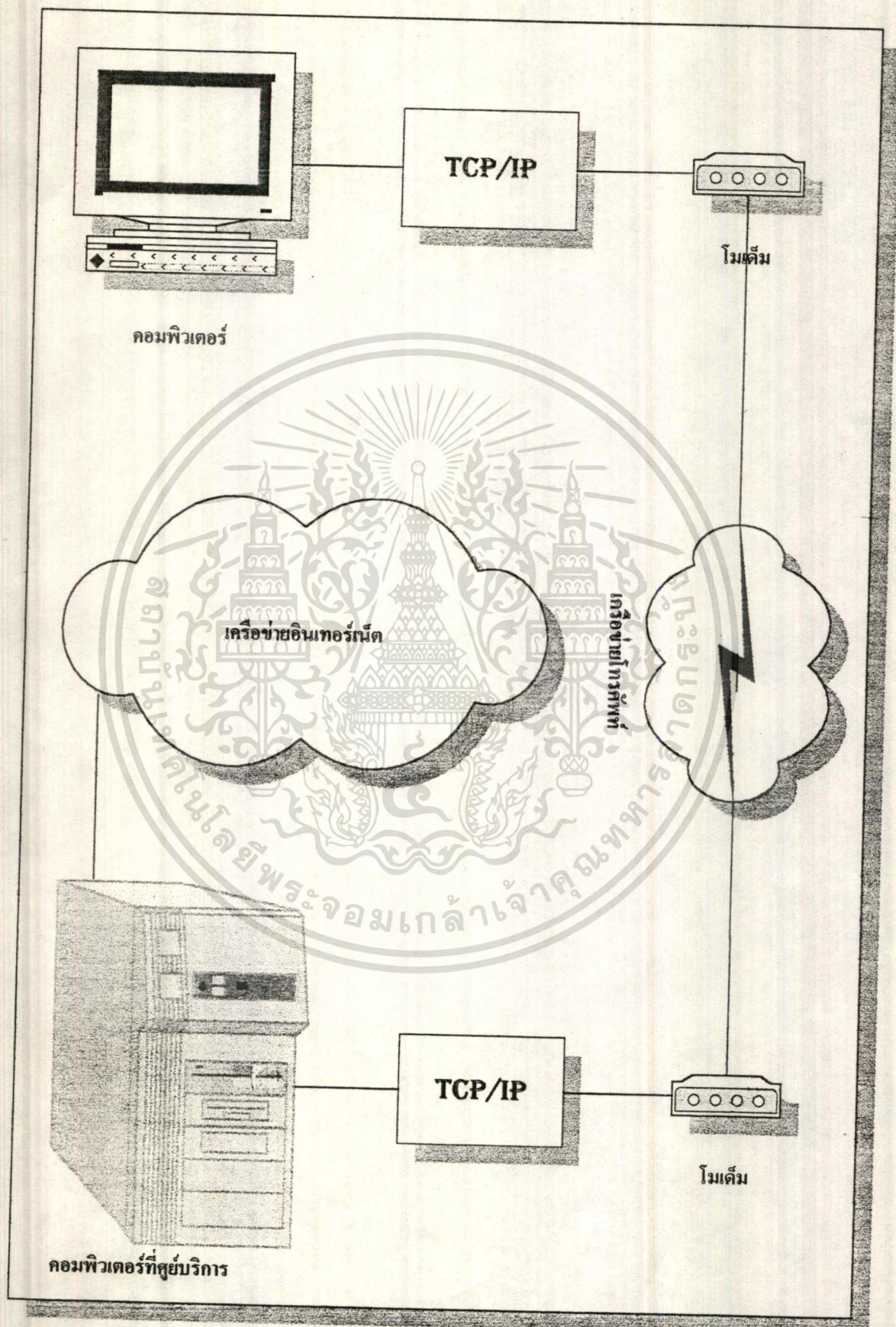
การเชื่อมต่อที่อนุญาตให้มีการใช้จอหลายชุดได้แก่การเชื่อมต่อโดยอาศัยโปรโตคอลเป็นตัวควบคุมโปรโตคอลที่ใช้งานอาจเป็น สลิป (SLIP Serial Link Internet Protocol), ซีลลิป (CSLIP Compression Serial Link Internet Protocol), พีพีพี (PPP Point to Point Protocol)

การใช้โปรโตคอลในการติดต่อระหว่างคอมพิวเตอร์ที่บ้านและคอมพิวเตอร์ที่ศูนย์บริการจะมีรูปแบบดังรูปที่แสดง การต่อเชื่อมแบบใช้โปรโตคอล ซึ่งหากมองโดยผิวเผินแล้วจะพบว่าไม่มีความแตกต่างทางด้านการจัดว่าอุปกรณ์ที่บ้านหรือสำนักงานในตอนแรกแต่อย่างใด เพราะยังคงใช้คอมพิวเตอร์ ไม่เต็มสายโทรศัพท์อยู่เช่นเดิม ส่วนที่แตกต่างได้แก่วิธีการติดตั้งซอฟต์แวร์และการใช้ซอฟต์แวร์ที่เพิ่มความซับซ้อนขึ้น ตลอดจนการเชื่อมต่อทางด้านปลายทางที่ศูนย์บริการซึ่งต้องมีการเตรียมพอร์ตเพื่อการเชื่อมต่อไว้ ในกรณีนี้ต้องติดต่อขอคำแนะนำจากผู้รับผิดชอบระบบเครือข่ายของศูนย์บริการนั้น จากนั้นก็จัดเตรียมอุปกรณ์ดังนี้

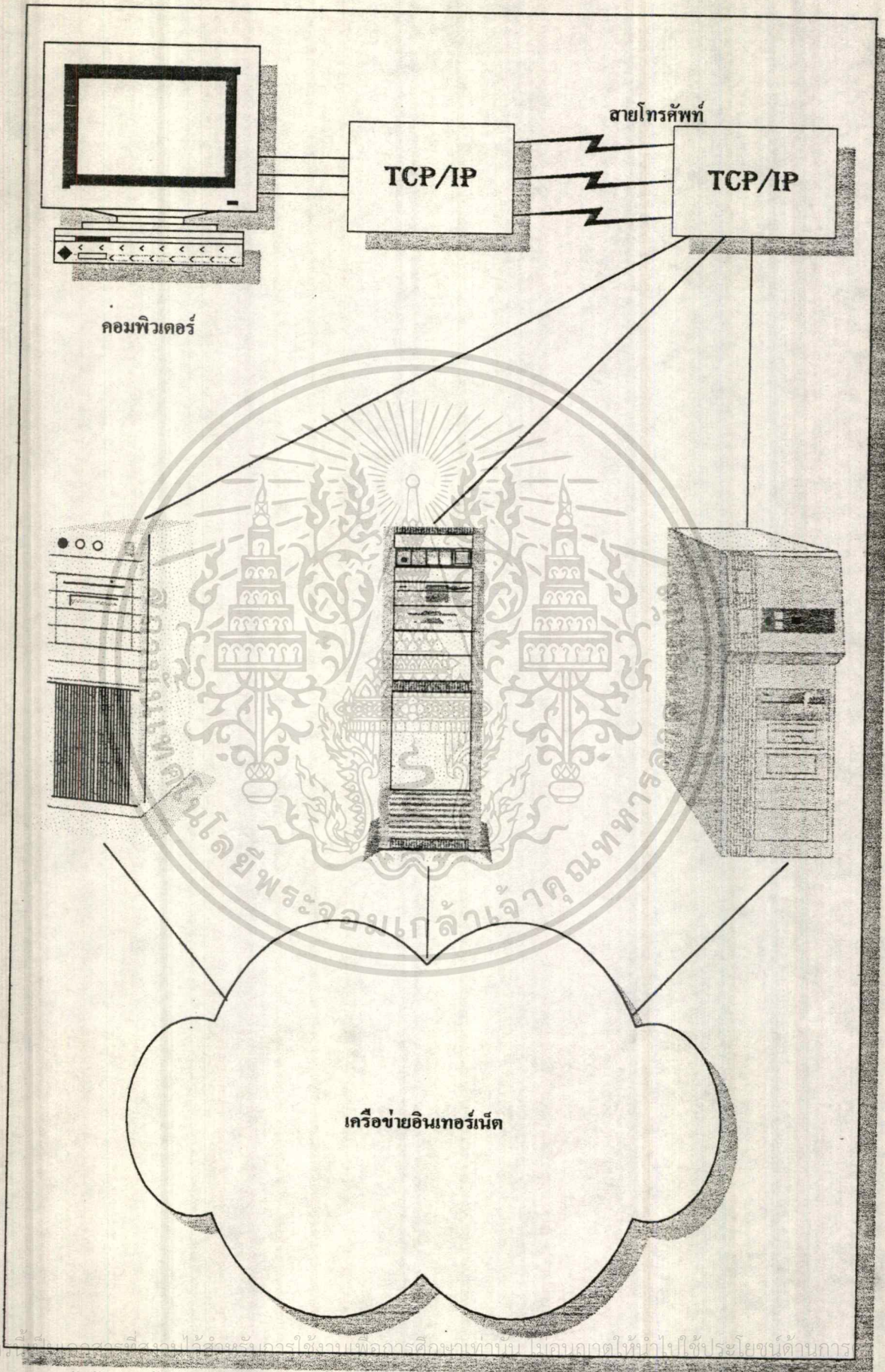
คอมพิวเตอร์ เลือกได้หลายชนิดเช่นเดียวกับการต่อเทอร์มินัล

โมเด็ม เช่นเดียวกันกับการต่อแบบเทอร์มินัล

ซอฟต์แวร์ ซอฟต์แวร์ที่จำเป็นต้องใช้งาน จะเป็นซอฟต์แวร์ที่มีความสามารถจัดการกับโปรโตคอลที่ซีพี/ไอพี ซึ่งมีทั้งซอฟต์แวร์ประสิทธิภาพสูงและจำหน่ายในเชิงการค้า อย่างเช่น พีซี-เอ็นเอฟเอส ของบริษัท ไมโครซิสเต็ม หรือซอฟต์แวร์สาธารณะที่มีให้ใช้โดยไม่ต้องเสียค่าใช้จ่าย เช่น เอ็นซีเอสเอ-เทลเน็ต , ดอสไอพี , พีซี-ไอพี หรือ เคอร์มิท เป็นต้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
การต่อเชื่อมโดยแบบใช้โปรโตคอล
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่ควรเผยแพร่ให้ผู้อื่นได้ใช้ประโยชน์ด้านการ

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิได้เผยแพร่ข้อมูลนี้ไปยังผู้อื่นโดยไม่ได้รับอนุญาตจากทางราชการ



ตัวกรอง

คุณอาจมองว่าคำสั่งเชลล์ส่วนมากมักจะเป็นตัวกรอง คือมีข้อมูลเข้าเพียงทางเดียวคือหน่วยรับข้อมูลมาตรฐาน กรณานิยาม ตารางเลขแฟ้ม คำสั่งเหล่านี้จะได้รับข้อมูลที่ละตัวอักษร และมีหน่วยแสดงผลอยู่ 2 หน่วย คือ หน่วยแสดงผลมาตรฐาน และหน่วยแสดงข้อผิดพลาดมาตรฐาน คำสั่งงานแต่ละคำสั่งจะกรองข้อมูลจากหน่วยรับข้อมูลมาตรฐาน หรือแก้ไขบางรูปแบบแล้วส่งข้อมูลไปยังหน่วยแสดงผลมาตรฐาน ขณะที่ข้อผิดพลาดจะส่งไปยังหน่วยแสดงข้อผิดพลาดมาตรฐาน แต่ข้อผิดพลาดมีโอกาสเกิดขึ้นน้อยเนื่องจากคำสั่งยูนิกซ์ส่วนใหญ่จะมีการปรับตัวให้เข้ากับค่าที่ส่งมาในสถานการณ์ต่างๆ เช่นถ้าไม่กำหนดให้แฟ้มใดเป็นส่วนรับข้อมูลมาตรฐานแล้ว เชลล์จะให้เครื่องปลายทางเป็นหน่วยรับข้อมูลมาตรฐาน สิ่งที่ยากที่สุดที่มักจะทำคือการใช้คำสั่ง cat (concatenate) ซึ่งเป็นคำสั่งที่ใช้ในการต่อเชื่อมสตริงนั่นเอง

ขอให้ท่านพิจารณารูป เครื่องปลายทางเป็นหน่วยรับ หน่วยแสดงผล และหน่วยรับข้อผิดพลาดมาตรฐาน เชลล์จะรอให้ป้อนข้อมูลจากจอภาพแล้วจึงแสดงผลออกมา สำหรับการหยุดทำงานให้กดแป้น break หรือแป้น delete หรือกด ctrl-d (EOF) อย่างใดอย่างหนึ่ง

คำสั่ง cat เป็นตัวกรองของเชลล์ที่ยากที่สุด ไม่มีการเปลี่ยนแปลงข้อมูล คือรับข้อมูล คือรับข้อมูลจากหน่วยรับมาตรฐานและแสดงผลไปยังหน่วยแสดงผลมาตรฐาน เมื่อดูแล้วเหมือนจะไม่มีประโยชน์อะไร แต่ถ้าต้องการดูแฟ้มบนเครื่องปลายทางต้องป้อนคำสั่ง cat ตามด้วยชื่อไฟล์ เชลล์จะเปิดแฟ้มและแสดงผลบนหน่วยแสดงผลมาตรฐาน คือเครื่องปลายทาง ข่าวสารเกี่ยวกับข้อผิดพลาดที่พบ เช่น ไม่มีแฟ้ม จะส่งไปยังหน่วยแสดงข้อผิดพลาดมาตรฐาน ซึ่งก็คือ เครื่องปลายทางนั่นเอง

ตัวกรองบางตัวคัดเลือกข้อมูลที่ต้องการดู เพิ่ม หรือเปลี่ยนข้อมูลตามการสั่งงานของผู้ใช้เช่นคำสั่ง grep(ทั่วไปจะใช้ตรวจสอบนิพจน์ทั่วไปและพิมพ์) จะหาสิ่งที่ปรากฏในคำหรือ วลีในแฟ้มของยูนิกซ์ เช่น คำสั่งต่อไปนี้จะหาคำว่า “ผู้แต่ง” ที่ปรากฏในแฟ้ม “บทที่หนึ่ง”

grep “ผู้แต่ง” บทที่หนึ่ง

ผู้แต่ง : Lowel Jay Arthur

King Arthur and the Knights of the Round Table

จากตัวอย่าง มี 2 บรรทัดที่พบคำว่า Arthur โดยคำสั่ง grep กรองบรรทัดอื่นๆ ออกไปจากแฟ้ม “บทที่หนึ่ง” โปรดสังเกตว่าคำสั่งเชลล์ทั้งหมดแยกความแตกต่างระหว่างตัวอักษรตัวใหญ่และตัวเล็ก ตัวอย่างต่อไปแสดงให้เห็นว่าเชลล์สามารถใช้คำสั่งแก้ไขและเพิ่มเติมข้อมูลได้ สมมติว่าต้องการเปลี่ยนทุกที่ที่มีคำว่า เชลล์ เป็นเชลล์ โดยใช้คำสั่ง sed (บรรณาธิกรกระแส หรือ สดริ่ม อิติเตอร์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไปใช้ประโยชน์อย่างอื่น สิ่งนี้หมายถึงการคัดลอก การแก้ไข และการแจกจ่ายสิ่งนี้โดยไม่ได้รับอนุญาต

การเปลี่ยนแปลงทิศทางข้อมูลเข้า/ออก

การเปลี่ยนแปลงทิศทางข้อมูลเข้าออกก่อให้เกิด

1. การสร้างแฟ้มข้อมูลใหม่
2. การเพิ่มข้อมูลในแฟ้ม
3. การใช้แฟ้มที่มีอยู่เป็นข้อมูลเข้าสำหรับเซลล์
4. การรวมข้อมูลที่เป็นผลลัพธ์จาก 2 กระแส
5. การใช้ส่วนหนึ่งของคำสั่งเซลล์เป็นข้อมูลเข้า

การเปลี่ยนแปลงทิศทางข้อมูลเข้าและออกนำมาใช้เปลี่ยนทิศทางของหน่วยรับข้อมูลหน่วยแสดงผล และหน่วยระบุข้อผิดพลาดมาตรฐาน หรือหมายเลขแฟ้มที่ผู้ใช้กำหนดเอง (ดังตารางแสดง หมายเลขแฟ้ม) ในขณะใดขณะหนึ่งสามารถเปิดแฟ้มได้ 20 แฟ้ม ใช้หมายเลขแฟ้มตั้งแต่ 0-19 โดยหมายเลข 0-2 กันไว้สำหรับหน่วยรับข้อมูลมาตรฐาน (0) หน่วยแสดงผลมาตรฐาน (2)

file-descriptor-1 operator file-name (e.g., 2> errorfile)

file-descriptor-1 operator file-descriptor-2 (e.g., 2>&1)

รูปแบบแรกเปิดแฟ้มข้อมูลเข้าหรือออก และกำหนดให้เป็นไปตามหมายเลขแฟ้มที่กำหนดให้แก่แฟ้มนั้น รูปแบบที่สองคัดลอกหรือกำหนดค่าหมายเลขแฟ้มหนึ่งให้กับหมายเลขแฟ้มอื่นหมายเลขหนึ่งเซลล์จะใช้ตัวปฏิบัติการจากตาราง ตัวปฏิบัติการเปลี่ยนทิศทาง

ตาราง ตัวปฏิบัติการเปลี่ยนทิศทาง

ตัวปฏิบัติการ	การกระทำ
<	เปิดแฟ้มที่ตามมาเป็นหน่วยรับข้อมูลมาตรฐาน
>	เปิดแฟ้มที่ตามมาเป็นหน่วยแสดงผลมาตรฐาน
>>	เพิ่มข้อมูลในแฟ้มที่ตามเครื่องหมายนี้มา
<<delimiter	กำหนดเป็นหน่วยรับข้อมูลมาตรฐานจนถึงตัวแบ่ง
m<&n	ใช้แฟ้มหมายเลข n เป็นข้อมูลเข้าของแฟ้มหมายเลข m เมื่อมีการเรียกใช้
m>&n	ต่อแฟ้มหมายเลข m ด้วยแฟ้มหมายเลข n
m>>&n	เพิ่มแฟ้มหมายเลข m เข้าไปยังแฟ้มหมายเลข n
!	ส่งผ่านหน่วยแสดงผลมาตรฐานไปเป็นหน่วยรับมาตรฐาน

ด้านการค้า

ตาราง การเปลี่ยนทิศทางข้อมูลเข้าออก

บอริ่น/คอรันเทิลด์	ซีเชลด์	การกระทำ
<file	<file	เพิ่มนี้เป็นหน่วยรับข้อมูลมาตรฐาน
>file	>file	ใส่หน่วยแสดงผลมาตรฐานลงในแฟ้ม
>>file	>>file	เพิ่มข้อมูลหรือสร้างแฟ้มใหม่ (ถ้ายังไม่มีแฟ้มนั้น)
<<delimiter	<<delimiter	ให้หน่วยรับมาตรฐานจากเซตคั่นถึงตัวแบ่ง
4<file		เปิดแฟ้มข้อมูลหมายเลข 4
<&4		อ่านข้อมูลจากแฟ้มหมายเลข 4
2<file		ใส่หน่วยประมวลผลมาตรฐานลงในแฟ้มใหม่
>&3		เขียนลงไปถึงแฟ้มหมายเลข 3
0&4		ไว้แฟ้มหมายเลข 4 เป็นหน่วยรับข้อมูล
2>&1	>&	รวมหน่วยแสดงผลมาตรฐานกับหน่วยประมวลผลมาตรฐาน
2>>&1	>>&	ต่อหน่วยแสดงผลมาตรฐานกับหน่วยประมวลผลมาตรฐาน
	>file	ข้าม noclobber และสร้างแฟ้มใหม่
	>>file	ข้าม noclobber และเพิ่มข้อมูลในแฟ้ม
<&	<&	ปิดแฟ้มหมายเลข 0 (หน่วยรับข้อมูลมาตรฐาน)
2>&		ปิดแฟ้มหมายเลข 2 (หน่วยประมวลผลมาตรฐาน)
2		ส่งหน่วยประมวลผลไปส่งคำสั่งต่อไป
2>&1	&	ส่งหน่วยประมวลผลและหน่วยรับข้อมูล และส่งต่อผลลัพธ์

ขอให้ท่านดูตัวอย่างจากการใช้คำสั่ง sed โดยคำสั่งนี้จะรู้ได้เองว่าแฟ้มที่เปิดอยู่ คือ “บทที่หนึ่ง” เป็นข้อมูลเข้า แต่จะเขียนเป็นคำสั่งได้ตามแบบดังต่อไปนี้

`sed -e "s/shell/Shell/g" < บทที่หนึ่ง > บทที่หนึ่งใหม่`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยระบุข้อมูลผิดพลาดมาตรฐานคืออะไร หน่วยนี้ส่งข้อมูลตรงไปยังเครื่องปลายทางหรือไม่ คำตอบคือ ใช้การเปลี่ยนทิศของหน่วยระบุข้อมูลผิดพลาดมาตรฐานมาเป็นแฟ้มในบางครั้งมีประโยชน์ในการตรวจหาข้อผิดพลาดในคำสั่งงานเชลล์ การเปลี่ยนทิศทางในบอร์นและคอร์นเชลล์เพื่อบอกว่าหน่วยแสดงข้อผิดพลาดมาตรฐาน ควรจะเก็บไว้ในแฟ้มดังตัวอย่างต่อไปนี้

```
sed -e "s/shell/Shell" บทที่หนึ่ง > บทที่หนึ่งใหม่ > แฟ้มข้อผิดพลาด
```

บางครั้งเราใช้ตัวปฏิบัติการเหล่านี้ในการรวมหน่วยแสดงผล และหน่วยระบุข้อมูลผิดพลาดมาตรฐานมาเป็นข้อมูลชุดเดียวและเก็บไว้ในแฟ้มเดียว ตามตัวอย่างต่อไปนี้

```
sed -e "s/shell/Shell/g" บทที่หนึ่ง 2>&1 > บทที่หนึ่งใหม่
```

```
sed -e "s/shell/Shell/g" บทที่หนึ่ง >& บทที่หนึ่งใหม่
```

ประโยค "2>&1" บอกเชลล์ให้กำหนดหน่วยระบุข้อมูลผิดพลาดมาตรฐาน (2) มีหมายเลขแฟ้มเดียวกับหน่วยรับข้อมูลมาตรฐาน (&1) และเชลล์จะเปลี่ยนทิศทางผลลัพธ์ลงในแฟ้ม บทที่หนึ่งใหม่ เชลล์มีคุณสมบัติพิเศษสองประการคือ ประการแรกสามารถทำการแก้ไขเพิ่มเติมในแฟ้มและใช้คำสั่งเชลล์เป็นข้อมูลเข้าเพื่อทำการควบคุมสถานการณ์พิเศษ

สัญลักษณ์เปลี่ยนทิศทางผลลัพธ์ ">" สร้างแฟ้มใหม่ถ้าไม่มีแฟ้มที่กำหนดในคำสั่งอยู่ก่อน แต่ถ้าหากมีจะเขียนทับแฟ้มที่มีอยู่แล้วนั้นเลย บางครั้งนำมาใช้ในการเขียนและเพิ่มข้อความลงในแฟ้มด้วยเครื่องหมาย ">>" ซึ่งเครื่องหมายนี้จะทำงานดังต่อไปนี้

1. ถ้าไม่มีแฟ้มที่กำหนดในคำสั่งนั้นอยู่จะสร้างแฟ้มใหม่ตามที่คำสั่งนั้นกำหนด
2. ถ้ามีแฟ้มตามที่กำหนดในคำสั่งนั้นอยู่แล้วจะเพิ่มข้อความใหม่ต่อเข้ากับข้อความเก่าที่มีอยู่เดิม

ตัวอย่างต่อไปจะเป็นเรื่องเกี่ยวกับการเขียนชุดคำสั่ง เชลล์เพื่อสร้างแฟ้มข้อมูลที่บอกข้อผิดพลาดและจะส่งข้อผิดพลาดนั้นไปยังผู้ที่สั่งให้ทำคำสั่งนั้นด้วย

```
echo "ข้อผิดพลาดครั้งแรก" > แฟ้มตัวอย่าง # สร้างแฟ้มเก็บข้อผิดพลาดใหม่
```

```
echo "ข้อผิดพลาดครั้งที่สอง" >> แฟ้มตัวอย่าง # เพิ่มข้อผิดพลาดในแฟ้มตัวอย่าง
```

```
echo "ข้อผิดพลาดครั้งที่สาม" >> แฟ้มตัวอย่าง # เพิ่มข้อผิดพลาดในแฟ้มตัวอย่างต่อไปอีก
```

```
mail เพื่อน < แฟ้มตัวอย่าง # ส่งแฟ้มตัวอย่างที่สร้างขึ้นไปให้เพื่อน
```

ข้อผิดพลาดจากตัวอย่างแรกเกิดขึ้น และเก็บไว้ในแฟ้มที่สร้างขึ้นชื่อ แฟ้มตัวอย่าง การใช้เครื่องหมาย ">>" จะช่วยสร้างแฟ้มใหม่ และหลีกเลี่ยงการเพิ่มข้อผิดพลาดใหม่ต่อจากข้อผิดพลาดเดิมในแฟ้มเดียวกันข้อผิดพลาดที่เกิดขึ้นๆ กันมาจะเพิ่มเข้าไปในแฟ้ม แฟ้มตัวอย่าง และในที่สุดก็จะส่งจดหมาย ไปยังผู้เรียกใช้คำสั่งการรวมข้อผิดพลาดเป็นข้อความเดียวเพื่อส่งจดหมายดีกว่าแยกเป็น 3 ข้อความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อ

stty - ตั้งตัวเลือกสำหรับเทอร์มินอล(terminal)

วิธีใช้

stty [-a | -g | options]

รายละเอียด

stty ใช้สำหรับรายงานหรือทำการตั้งค่าของเทอร์มินอลที่เป็นทางเลือกของ/O สำหรับอุปกรณ์ซึ่งใช้เป็นอินพุตปัจจุบัน ซึ่งคำสั่งนี้มี 4 รูปแบบ

stty	เพื่อรายงานการตั้งค่าที่ถูกตั้ง
stty -a	รายงานการตั้งค่าทุกอย่าง
stty -g	รายงานเป็นรูปแบบที่ใช้กับการตั้งค่า stty อื่นได้
stty options	ตั้งค่าตามตัวเลือกที่ถูกเลือก

ตัวเลือกที่ถูกตั้งจะเป็นการนำไปรวมกับกลุ่มของโหมด(modes group)ซึ่งเป็นตัวเลือกที่ถูกตั้งค่ามาก่อนหน้านี้ โดยตัวเลือกจะมีดังนี้

โหมดควบคุม

เลขของแถว	ตั้งค่าแถวของวินโดวในเทอร์มินอลเท่ากับเลขที่กำหนด
เลขของหลัก	ตั้งค่าหลักของวินโดวในเทอร์มินอลเท่ากับเลขที่กำหนด(ความกว้าง)

โหมดอินพุต

ignbrk (-ignbrk)	สนใจ(หรือไม่สนใจ)การหยุด(break)ที่เข้ามาทางอินพุต
brkint (-brkint)	ส่งสัญญาณ(Signal) (หรือไม่ส่งสัญญาณ) INTR ในการหยุด(break)
inlcr (-inlcr)	แมป(map) (ไม่แมป) การขึ้นบรรทัดใหม่เมื่อมีอินพุต CR (carriage return)เข้ามา
igncr (-igncr)	ยกเลิก (ไม่ยกเลิก) เมื่อมี CR เข้ามา
iuclc (-iuclc)	แมป(map)(ไม่แมป)อักขระตัวใหญ่ไปเป็นอักขระตัวเล็กเมื่อมีอินพุต
ixon (-ixon)	มี(ไม่มี)การควบคุม START/STOP (เอาท์พุทจะถูกหยุดเมื่อมีการส่งสัญญาณ ASCII DC3 และ จะถูกเริ่มต้นเมื่อส่งสัญญาณ ASCII DC1
ixany (-ixany)	อนุญาตให้ DC1 ส่งเอาท์พุททำการเริ่มต้นใหม่อีกครั้ง

ixoff (-ixoff)

ขอให้ระบบส่ง(ไม่ส่ง) อักขระ START/STOP เมื่อคิวของอินพุตใกล้จะว่างหรือ เต็ม

โหมดเอาท์พุท

opost (-opost)	ส่งเอาท์พุทโปรเซส(process)(ไม่ส่ง)ที่ยกเลิกทุกโหมดเอาท์พุท
olcuc (-olcuc)	แมป(map) (ไม่แมป) ตัวอักขระตัวเล็กไปเป็นอักขระตัวใหญ่
onocr (-onocr)	ไม่ทำ(ทำ) เอาท์พุทของ CR ที่หลักที่ 0
cr0 cr1 cr2 cr3	เลือกรูปแบบของการหน่วงเวลาเมื่อใช้ CR
nl0 nl1	เลือกรูปแบบของการหน่วงเวลาเมื่อใช้อักขระขึ้นบรรทัดใหม่
tab0 tab1 tab2 tab3	เลือกรูปแบบของการหน่วงเวลาเมื่อใช้ แทบ(tab)
bs0 bs1	เลือกรูปแบบของการหน่วงเวลาเมื่อใช้ปุ่มกดถอยหลัง(back space)
ff0 ff1	เลือกรูปแบบของการหน่วงเวลาเมื่อทำการเลื่อนหน้า
	โหมดโลคอล(local)
isig (-isig)	ให้(ไม่ให้)มีการตรวจสอบอักขระที่ต่อต้านกับการควบคุมพิเศษ INTR และ QUIT
icanon (-icanon)	ให้(ไม่ให้)มีการทำ ERASE และ KILL
echo (-echo)	เอคโค(echo)(ไม่เอคโค)ทุกๆอักขระที่ถูกพิมพ์
echoe (-echoe)	เอคโค(echo)(ไม่เอคโค) อักขระ ERASE ที่เป็นสายอักขระ ถอย หลัง-ที่ว่าง-ถอยหลัง(backspace-space-backspace)
echok (-echok)	เอคโค(ไม่เอคโค)อักขระบรรทัดใหม่หลังอักขระ KILL
echonl (-echonl)	เอคโค(ไม่เอคโค)อักขระบรรทัดใหม่

การจัดตั้งการควบคุม(Control Assignments)

control-character c ตั้งอักขระควบคุมไปยัง c ซึ่งสามารถควบคุม

อีราส(erase), คิล(kill), intr, ควิต(quit), eof, eol, min, หรือ ไทม์(time)(min และ ไทม์ใช้กับ -icannon และสำหรับ

ระบบที่รองรับการควบคุมงาน อักขระ susp และ dsusp ก็สามารถจัดตั้งได้เช่นกันสำหรับระบบที่รองรับการเปลี่ยนไปยังเชลล์(shell) การเปลี่ยน(switch) ก็สามารถจัดตั้งได้เช่นกัน ถ้า c เริ่มต้นด้วยเครื่องหมายรูปปหมวก(^)ค่าที่ถูกใช้จะเป็นอักขระที่ปุ่มคอนโทรล(Ctrl) เช่น ^d หมายถึง CTRL-d ^? หมายถึง DEL และ ^ หมายถึงไม่ได้กำหนด

โหมดคอมบายนะชัน(combination modes)

raw (-raw or cooked) ให้ (ไม่ให้) อินพุทและเอาท์พุท รอร์(raw) (ไม่มี ERASE, KILL, INTR, QUIT, EOT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

nl (-nl)	ตั้งค่า(ไม่ตั้ง) icrnl และ onlcr
lcase (-lcase)	ตั้งค่า(ไม่ตั้งค่า) xcase, iuclic, และ olcuc.
LCASE (-LCASE)	เหมือนกับ lcase (-lcase).
ek	ไม่ตั้งค่าอักขระ ERASE และ KILL โดยกลับไปเป็น ค่าดั้งเดิม # and @
sane	ไม่ตั้งค่าทุกโหมดและตั้งเป็นโหมดที่สมควร
term	ตั้งทุกโหมดให้เข้ากับปุ่มของเทอร์มินอลนั้น เมื่อเทอร์มินอลนั้นมีเทอม(term)ของ tty33, tty37,



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การINTERFACEกับแฟ้มข้อมูลยูนิกซ์

ในหัวข้อที่จะกล่าวต่อไปนี้จะเกี่ยวข้องกับINTERFACEระหว่างภาษาCและระบบปฏิบัติการยูนิกซ์โดยจะเน้นในหัวข้อของ อินพุทเอาต์พุทระดับต่ำ (Low Level I/O) คือคำสั่ง Read , Write และคำสั่งอื่นๆเช่น Open , Close ที่ใช้ในโครงงานชิ้นนี้

คำสั่งอินพุทเอาต์พุทระดับต่ำ(Low Level I/O) - Read และ Write

ในระดับขั้นที่ต่ำสุดของอินพุทและเอาต์พุทในยูนิกซ์ไม่มีการทำบัฟเฟอร์(buffering)หรือบริการอื่น การทำI/Oจะการทำผ่านฟังก์ชันที่เรียกว่า Read และ write สำหรับทั้งคู่argumentแรกคือfile descriptor ส่วน argumentที่สองเป็นบัฟเฟอร์ของโปรแกรมที่ข้อมูล(data)จะมาหรือไปและ argumentที่สามคือจำนวนของไบต์(byte)ที่ถูกส่งดังนี้

```
n_read = read(fd,buf,n);
```

```
n_write= read(fd,buf,n);
```

การเรียกฟังก์ชันแต่ละครั้งจะส่งกลับ(return)ไบต์ที่ถูกส่งจริง ในการ read เลขจำนวนไบต์ที่ถูกส่งกลับอาจน้อยกว่าจำนวนที่ถูกขอ ค่าส่งกลับที่เป็นศูนย์นั้นแสดงถึงสิ้นสุดแฟ้มข้อมูล(end of file)และ -1ชี้ว่ามีความผิดพลาดเกิดขึ้น สำหรับการ write ค่าที่ถูกส่งกลับเป็นเลขจำนวนของไบต์ที่อ่านได้จริงๆ

เลขจำนวนของไบต์ที่จะ read หรือ write สามารถกำหนดได้ตรงๆ ค่าที่มักใช้กันมีสองค่าคือ 1 หมายความว่าหนึ่งตัวอักษรในหนึ่งครั้งหรือไม่มีบัฟเฟอร์(unbuffered)และ 512 ซึ่งพ้องกับขนาดบล็อก(block)ทางกายภาพในอุปกรณ์ต่างๆ

นำสิ่งต่างๆมารวมกันเราสามารถเขียนโปรแกรมที่จะลอก(copy)อินพุทของมันไปยังเอาต์พุทของมันดังนี้

```
#define BUFSIZE 512 /* เป็นขนาดที่เหมาะสมที่สุดสำหรับ PDP-11 UNIX*/
main() /* ลอก อินพุท ไปยัง เอาต์พุท */
{
    char buf[BUFSIZE];

    int n;

    while ((n=read(0,buf,BUFSIZE)) > 0 )
        write(1,buf,n);
}
```

ชุดคำสั่งของ CTYPE.H คำสั่งแยกแยะอักขระ(character test Macros)

isalpha(c)	อักขระ : a-z A-Z
isupper(c)	อักขระตัวใหญ่ : A-Z
islower(c)	อักขระตัวเล็ก : a-z
isdigit(c)	ตัวเลข : 0-9
isxdigit(c)	เลขฐานสิบหก : 0-9 a-f A-F
isalnum(c)	ตัวอักษร หรือ ตัวเลข
isspace(c)	ช่องว่าง ,แท็บ(tab) ,บรรทัดใหม่,เวอริทิคอลแท็บ(vertical tab) ,เลื่อนหน้า,ส่งกลับ(return)
ispunct(c)	ไม่ใช่ตัวอักษร หรือ อักขระควบคุม หรือ ช่องว่าง
isprint(c)	สามารถพิมพ์ได้หรือไม่
iscntrl(c)	อักขระควบคุม : $0 \leq c < 040$ $c == 0177$
isacii(c)	อักขระแบบ ASCII : $0 \leq c \leq 0177$

คำสั่งที่ทำงานกับสายอักขระ

strcat(s,t)	ต่อสายอักขระ t ไปยังสายอักขระ s ; ส่งกลับ s
strncat(s,t,n)	ต่อสายอักขระ t ไปยังสายอักขระ s เป็นจำนวน n ตัว
strcpy(s,t)	ลอก t ไปยัง s ; ส่งกลับ s
strncpy(s,t,n)	ทำการลอกเป็นจำนวน n อักขระ
strcmp(s,t)	เปรียบเทียบ s กับ t , ส่งกลับ <0 ,0, >0 สำหรับ < ,== ,>
strncmp(s,t,n)	เปรียบเทียบเป็นจำนวน n อักขระ
strlen(s)	ส่งกลับความยาวของ s
strchr(s,c)	ส่งกลับเป็น พอยเตอร์(pointer)ไปยังอักขระแรกของสายอักขระ s
strchr(s,c)	ส่งกลับเป็น พอยเตอร์(pointer)ไปยังอักขระสุดท้ายของสายอักขระ s
atoi(s)	ส่งกลับเลขอินที่เจอร์(integer)ของ s
atof(s)	ส่งกลับเลขโฟลตติง พ้อยท์(floating point)ของ s
malloc(n)	ส่งกลับเป็น พอยเตอร์ของหน่วยความจำเป็นจำนวน n byte calloc(n,m)ส่งกลับเป็น พอยเตอร์ของหน่วยความจำเป็นจำนวน n*m byte
	โดยทั้ง malloc และ calloc จะส่งกลับ char *
free(p)	ปลดปล่อยการจองของทั้ง malloc และ calloc

บางส่วนของชุดคำสั่ง STDIO.Hที่มักนำมาใช้

stdin	สแตนดาร์ด อินพุท (standard input)
stdout	สแตนดาร์ด เอาท์พุท (standard output)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

stderr	สแตนด์อาร์ท เออร์เรอร์ (standard error)
EOF	จบเพิ่มข้อมูล ซึ่งปกติจะเป็น -1
NULL	ไม่มีพอยเตอร์ ซึ่งปกติจะเป็น 0
FILE	ถูกใช้เพื่อประกาศพอยเตอร์ของเพิ่มข้อมูล
BUFFER	ขนาดของ บัฟเฟอร์(buffer)ของI/Oโดยปกติ(มักเป็น 512 หรือ 1024)
getc(fp)	ส่งกลับอักขระ 1 ตัวจากสาย(stream)ของ fp
getchar()	getc(stdin)
putc(c,fp)	ใส่อักขระ c ลงบนสาย(stream)ของ fp
putchar(c)	putc(c,stdout)
feof(fp)	ไม่เป็น 0 เมื่อจบเพิ่มข้อมูลของสาย(stream)ของ fp
ferror(fp)	ไม่เป็น 0 เมื่อเกิดความผิดพลาดบนสาย(stream)ของ fp
fileno(fp)	บรรยายเพิ่มข้อมูลสำหรับสายของ fp
fp=fopen(s,mode)	เปิดเพิ่มข้อมูล s โหมด(mode) "r","w","a" สำหรับ อ่าน,เขียน, ต่อ(ส่งกลับ NULL ถ้าเกิดความผิดพลาด)
c=getc(fp)	รับอักขระ
putc(c,fp)	ส่งอักขระ
ungetc(c,fp)	ส่งอักขระกลับเข้าไปในเพิ่มข้อมูลเข้า fp ซึ่ง 1 อักขระจะถูกส่งกลับต่อการเรียก ฟังก์ชันหนึ่งครั้ง
scanf(fmt,a1,...)	อ่านอักขระจาก stdin มาถึง a1 ซึ่งแต่ละ ai ต้องเป็นพอยเตอร์
fscanf(fp,...)	อ่านจากเพิ่มข้อมูล fp
sscanf(s,...)	อ่านจากสายอักขร s
printf(fmt,a1,...)	รูปแบบ a1 ตาม fmt พิมพ์บน stdout
fprintf(fp,...)	พิมพ์ ... ลงเพิ่มข้อมูล fp
sprintf(s,...)	พิมพ์ ... ลงในสายอักขร s
fgets(s,n,fp)	อ่านอักขระ n ตัวจาก fp
fputs(s,fp)	พิมพ์สายอักขร s บนเพิ่มข้อมูล fp
fclose(fp)	ปิดเพิ่มข้อมูล fp
fp=popen(s,mode)	เป็นไปท์(pipe)ไปยังคำสั่ง s
pclose(fp)	ปิดไปท์ fp
system(s)	ทำงานตามคำสั่ง s และระจกว่าเสร็จสมบูรณ์

ฟังก์ชันที่ใช้สำหรับการหน้าจอ

curses เป็นยูนิกซ์คลังของฟังก์ชัน ของฟังก์ชันที่ใช้ควบคุม ส่วนแสดงผลทางจอภาพโดยการใช้โปรแกรมที่เขียนด้วยภาษาซี โดย *curses* จะจัดหาไดรเวอร์ของจอภาพ สำหรับโปรแกรม หรือจะช่วยโปรแกรมเมอร์ในการพัฒนาการทำงานร่วมกันกับผู้ใช้

แนวคิดเกี่ยวกับวินโดว

curses เป็น คลังของฟังก์ชันในยูนิกซ์ซึ่งคลังของฟังก์ชันนี้มีใช้ในภาษาซี ซึ่งการในไปใช้นั้นสามารถใช้ *curses* ได้โดยไม่คำนึงถึงชนิดของ เทอร์มินัล โดยฟังก์ชันต่างๆ ของ *curses* มีดังนี้ :

1. เคลื่อน cursor ไปยังตำแหน่งใดๆ บนจอภาพ
2. แทรกตัวอักษร หรือทำ ตัวเน้น บนตำแหน่งใดๆ ก็ได้บนจอ
3. วาดกรอบรอบๆ วินโดว ด้วยตัวอักขระที่ท่านเลือกไว้
4. จัดการวินโดวได้อย่างอิสระ
5. แบ่งส่วนแสดงผลทางจอภาพ เป็นพื้นที่สี่เหลี่ยมเล็กๆ

stdscr หมายถึง standard screen ซึ่งหมายถึง วินโดว หรือเซตของวินโดวที่ปรากฏอยู่บนจอภาพในขณะนั้น และคำว่า screen จะหมายถึงคุณสมบัติทางด้านกายภาพของจอภาพนั้น ส่วนวินโดว ก็คืออาร์เรย์ของอักขระ ที่จัดการกับส่วนหนึ่ง หรือทั้งหมดของจอภาพ

ในตอนเริ่มแรก จอภาพจะถูกเติมด้วย ตัวอักขระ ช่องว่าง ทั้งจอภาพจนเต็ม ขณะที่คุณใช้ *curses* จะเป็นการทำงานโดยนำตัวอักขระที่ต้องการไปวางไว้ในตำแหน่งที่ต้องการบนจอภาพ โดยการทำการเปลี่ยนแปลง บิทให้เป็น เปิด หรือ ปิด ใน หน่วยความจำที่เก็บสถานะของบิทเหล่านั้น โดยมีคำสั่งของฟังก์ชันใน *curses* ที่ใช้ในการทำงานต่างๆ โดยจะยกเฉพาะที่ใช้กันบ่อยครั้งให้ท่านทราบเสียก่อน ฟังก์ชันต่างๆ ดังกล่าวนั้นมีดังต่อไปนี้ :

```

insch()
addch()
addstr()
printw()

```

ในการเลื่อน logical pointer ของจอภาพ โดยทั่วไปจะทำแบบเรียงตามลำดับ คือเคอร์เซอร์ จะเป็นตัวกำหนดตำแหน่งของตัวอักขระที่จะถูกพิมพ์เป็นตัวถัดไป แต่จุดประสงค์ของวินโดวจะจัดหาโอ/ไอแบบบนอนซีแควนเขียน เพื่อที่จะทำสิ่งเหล่านี้แล้ว *curses* จะทำงานกับ logical cursor แทนที่จะทำงานกับ physical cursor โดยที่ logical cursor คือ (y, x) ที่ระบุตำแหน่งของทุกๆวินโดว ซึ่ง y และ x นั้น จะเป็นตัวแปรซึ่งค่าของตัวแปรจะกำหนดตำแหน่งของ cursor ที่จะปรากฏบนภาพที่แสดงในจอ คำสั่ง move(y, x) ใช้ในการกำหนดตำแหน่งของ logical cursor ก่อนที่จะ แทรก หรือลบตัวอักขระ ที่กำหนดไว้ในตำแหน่งนั้น

การ Refresh จอภาพ หลังจากได้แทรกตัวอักษรเข้าไปในอาร์เรย์ของอักขระ คุณจะต้องทำการส่งอาร์เรย์ของอักขระ นั้นออกไปแสดงผลทางจอภาพ นั่นก็คือการที่คุณทำการเขียนจากอาร์เรย์ของอักขระไปยังจอภาพ ถ้าเทอร์มินัล เป็นแบบบิตแมต การ เขียนจอภาพใหม่ จะทำเองโดยอัตโนมัติ โดยทุกครั้งที่ electron gun กวาดผ่านจอภาพที่ตำแหน่งใดๆ ก็จะได้แสดงผลของรูปที่เป็นบิตแมตในขณะนั้นออกมาแสดงโดยทันที แต่สำหรับ curses แล้วจอภาพจะถูก refresh นั้นเมื่อท่านได้ใช้คำสั่งเขียนจอภาพใหม่เท่านั้น แต่การที่ curses ทำเช่นนี้ (ต้อง refresh จอภาพทุกครั้งที่จะให้แสดงผลการเปลี่ยนแปลง) เพราะสามารถทำการเปลี่ยนแปลงต่างๆ ที่ต้องการให้เห็นว่าปรากฏพร้อมกันได้เนื่องจากการเปลี่ยนแปลงที่ทำเป็นขั้นตอนนั้นจะไม่ปรากฏออกมาจนถึงจุดที่โปรแกรมเมอร์ ต้องการให้ปรากฏจึงกำหนดการ refresh นั้น และคำสั่ง refresh จะช่วยคำนวณหาทางที่ดีที่สุดใน การเปลี่ยนแปลงการแสดงผลทางจอภาพซึ่งจะทำให้การเปลี่ยนแปลงเป็นไปได้อย่างรวดเร็วและเป็นไปอย่างต่อเนื่องอีกด้วย ขณะนี้เราจะทำการสรุปการทำงานของ วินโดว์ เป็นขั้นตอนดังต่อไปนี้

Initialize วินโดว์ เพื่อแสดงวินโดว์ บนจอภาพ,

||

เพิ่มหรือย้ายอักขระเข้าไปยัง อาร์เรย์ของอักขระ,

||

Refresh เพื่อแสดงอักขระที่มีการเปลี่ยนแปลงของวินโดว์ บนจอภาพ,

||

และ refresh อีกครั้งที่จอภาพเพื่อให้การเปลี่ยนแปลงเกิดบนจอภาพ.

เราขอเน้นว่าเป็นเรื่องที่จะเป็นมากที่คุณจะต้องเข้าใจ อัลกอริทึม refresh() และขอให้คุณเข้าใจด้วยว่าการ refresh นั้นเพราะถ้าทำเช่นนั้นแล้วจะทำให้ทำงานที่ 9600 กิโลไบต์ต่อวินาที ได้อย่างมีประสิทธิภาพ และ จะทำงานได้อย่างไม่สะดุดเป็นอย่างมากกับการทำงานที่ 1200 กิโลไบต์ต่อวินาที แต่การทำงานของ refresh() จะพยายามทำการเปลี่ยนแปลงส่วนที่เป็นส่วนต่างกับสิ่งที่ปรากฏบนจอปัจจุบันเท่านั้น (เป็นการเปลี่ยนแปลงให้น้อยที่สุดเท่าที่จำเป็นเท่านั้น) จะทำให้การทำงานที่ความเร็วไม่สูงนักเป็นไปได้อย่างมีประสิทธิภาพมากขึ้น

curscr คือ จอภาพที่ปรากฏอยู่ ณ ปัจจุบัน โดย *curses* ไม่สามารถจะทราบจอภาพในปัจจุบันนั้นแสดงอะไรอยู่โดย *curses* จะต้องหาว่ามี ตัวอักขระ อะไรที่แสดงอยู่บนจอภาพในขณะนี้บ้าง และ ตัวอักขระ เหล่านั้นอยู่ที่ตำแหน่งใดของจอภาพ แต่การหาตัวอักขระที่ละตัวนั้นเป็นทำให้งานดำเนินไปได้อย่างเชื่องช้า ฉะนั้น *curses* จึงเก็บ รูปภาพ ของจอภาพนั้นไว้ให้เห็นสิ่งที่ปรากฏใน วินโดว์ ซึ่งภาพ ที่เก็บไว้นั้นเรียกว่า *curscr* โดยที่ *curscr* นั้นจะเหมือนกับ *stdscr* ซึ่งจะถูกสร้างขึ้นโดยอัตโนมัติขณะเริ่มการทำงาน

ของ *curses* ด้วยคำสั่ง *initscr()* *curscr* เป็น วินโดว์ ซึ่งมีขนาดเท่ากับขนาดของจอภาพทางกายภาพ เมื่อ *refresh()* ถูกเรียกจะทำการเขียนตัวอักษรไปยัง เทอร์มินัล โดยที่ *curscr* บรรจุนามของจอภาพเช่นเดียวกับ *curses* คำสั่ง *refresh()* เป็นเพียงฟังก์ชันที่ทำการเปลี่ยนแปลงภาพที่ปรากฏบนจอ ของ *curses*

การทำงานกับหลายวินโดว์

คุณสามารถทำงานดังมากมายโดยใช้เพียงวินโดว์ *stdscr* บางครั้ง *stdscr* ก็ไม่เป็นที่เพียงพอในงานต่างๆบางอย่างด้วย เช่นงานที่ต้องการใช้แสดงข้อความข่าวสารต่างๆ มากกว่าหนึ่งส่วน ซึ่งเราสามารถจะสร้างหน้าต่างวินโดว์ได้สองวิธี ดังนี้:

1. แบ่งวินโดว์ที่มีอยู่เป็น วินโดว์ย่อย (sub-windows) ด้วยคำสั่ง *subwin()*
2. สร้าง วินโดว์ ด้วยการ กำหนด วินโดว์ ขึ้นมาใหม่ ด้วยคำสั่ง *newwin()*

คุณสามารถทำงานต่างๆ ได้อย่างมากมายด้วยวินโดว์หลายๆหน้าต่าง แต่อาจเกิดความสับสนขึ้นบางอย่างในการทำงาน ขอให้คุณยึดหลักต่างๆ เหล่านี้ขณะท่านปฏิบัติงานเมื่อทำงานกับวินโดว์แบบหลายๆหน้าต่าง

1. ทำการออกแบบว่า จะวางวินโดว์ ได้ไว้ที่ไหน
2. จำไว้ว่าวินโดว์แต่ละอันเป็นอิสระต่อกันมีจุดประสงค์การทำงานต่างกัน
3. ไม่ควรลืมว่าวินโดว์ย่อยจะใช้ อาร์เรย์ของอักขระเดียวกันกับวินโดว์พ่อ
4. ทำความเข้าใจกับ คำสั่ง *refresh* และ *curscr* เพราะคำสั่งเหล่านี้จะทำการกับ วินโดว์ เดียวที่ท่านได้กำหนดว่าจะทำงานด้วยไว้

โครงสร้างโดยรวมของโปรแกรม *curses*

สำหรับ *curses* แล้วทุกๆโปรแกรม มีโครงสร้างพื้นฐานเหมือนๆ กัน ขอให้ท่านดูตัวอย่างโปรแกรมข้างล่างนี้ประกอบตามไปด้วย

```
#include <curses.h>

main()
{
    initscr();
    endwin();
    exit(0);
}
```

เราจะทำการพิจารณาไปที่ละบรรทัด เริ่มด้วย header file `curses.h` ต้องทำการรวม header file อันนี้ ทุกๆ โปรแกรมที่มีการใช้พื้นที่ของ `curses` ภายใต้ System V, `curses.h` จะถูกรวมเข้าไปด้วยอัตโนมัติเมื่อท่านได้ทำการรวม file header ชื่อ `termio.h` ซึ่งจะมีโครงสร้างข้อมูล และมาโคร ที่สำคัญๆ ของ `curses` บรรจุอยู่ด้วย

ในการใช้งาน `curses` นั้นจะต้องเรียกใช้คำสั่ง `initscr` ก่อนเสมอโดยคำสั่งมีรูปแบบดังนี้ `initscr()`; เพื่อจะทำการ initialize โครงสร้างข้อมูลที่บรรจุใน `curses` และเพื่อเป็นการตรวจสอบชนิดของเทอร์มินัล จากตัวแปรชื่อ `term` เช่นเดียวกับคำสั่ง `endwin` ซึ่งต้องถูกเรียกทุกครั้งเช่นกันก่อนที่จะทำการออกจากการทำงานของโปรแกรม นั้นๆ โดยคำสั่ง `endwin` นั้นจะ นำค่าเก่าที่เก็บไว้มาเป็นค่าเริ่มต้นเดิมก่อนเข้าทำงานในโปรแกรม state ของเทอร์มินัล และทำการเลื่อนตำแหน่งของ cursor ไปยังมุมล่างซ้ายของจอภาพอีกด้วย

ขอให้ท่านดูการใช้งานคำสั่ง `refresh` จากตัวอย่างโปรแกรมดังต่อไปนี้

```
#include <curses.h>
main()
{
    /* main body */
    initscr();
    refresh();
    endwin();
}
```

คำสั่ง `refresh` หรือที่ใช้กันทั่วไปคือคำสั่ง `wrefresh` ใช้เมื่อต้องการให้โปรแกรมแสดงการเปลี่ยนแปลงที่เกิดขึ้นนั้นแสดงออกทางจอภาพ นั่นก็คือการทำให้ physical screen แสดงภาพออกมาให้เหมือนกันกับรูป logical ของโปรแกรมโดยคำสั่ง `refresh` นั้นจะทำการคัดลอกสิ่งที่อยู่ในวินโดว์ (`stdscr`) ให้ไปปรากฏบน screen ส่วน `wrefresh` นั้นทำเช่นเดียวกับ `refresh` สำหรับ วินโดว์ ที่ได้ถูกกำหนดว่าเป็นวินโดว์ ณ เวลาปัจจุบัน ในกรณีของตัวอย่างนี้ ไม่มีข้อมูลใดๆ ที่ถูกเขียนโดยคำสั่งในโปรแกรมดังนั้นจอภาพก็จะยังคงว่างเช่นเดิมนั่นเอง สำหรับ `curses` นั้นเป็น คลังของฟังก์ชัน ที่มีฟังก์ชันจำนวนมากมายและผู้ใช้ท่านนั้นสามารถใช้งาน `curses` ได้ไม่ยากเย็นอะไรนักอีกด้วย

ชื่อ

```
curses()
```

วิธีใช้ และการคอมไพล์(compile)

```
#include <curses.h
```

```
cc [flags] file ... -lcurses [libraries]
```

คำบรรยาย

หน้าที่ประจำของcursesคือทำการกระทำหน้าจодด้วยวิธีที่สะดวก การเริ่มต้นใช้cursesต้องใช้ `initscr()` ซึ่งต้องเรียกใช้ก่อนหน้าที่จะเรียกใช้ส่วนทำงาน(routine)อื่นๆทั้งหมดซึ่งเกี่ยวข้องกับ วินโดว์(windows)และ หน้าจอ(screen)

`endwin()` ควรใช้ก่อนที่จะทำการออก และเพื่อรับอักขรต่อครั้งที่เข้ามา(charater-at-a-time input)โดยไม่มี การสะท้อนกลับ(echoing)ซึ่งโปรแกรมแบบที่อิงหน้าจอ(screen oriented)มักเรียกใช้:

```
nonl(); cbreak(); noecho();
```

โดยลักษณะแล้วการกระทำกับข้อมูลที่เราเรียกว่า “วินโดว์ (windows)” ซึ่งสามารถอธิบายได้ว่าเป็นตาราง สองมิติของตัวอักษรซึ่งแทนทุกส่วนของหน้าจอ CRT windowที่กำหนดไว้ดั้งเดิมเรียกกันว่า `stdscr` และ อื่นๆสามารถกำหนดโดยใช้ `newwin`

Windows หมายความว่า การตั้งค่าตัวแปร WINDOW * และ WINDOW ได้ถูกกำหนดไว้ใน<curses.h>ซึ่งเป็นโครงสร้างของภาษา C ซึ่งโครงสร้างทางข้อมูลถูกเปลี่ยนแปลงเพื่อการใช้งานโดยใช้ฟังก์ชัน(function) ต่างๆซึ่งจะอธิบายต่อไป ในพื้นที่ที่มักใช้กันมากคือ `move()`,`addch()` จากนั้น `refresh()` จะถูกเรียกซึ่งจะบอกให้ ทำการเขียนหน้าจอของผู้ใช้ใหม่ Mini-Curses

ฟังก์ชัน (Functions)

`addch(ch)*`

เติมอักขระสู่stdscrเหมือนคำสั่ง `putchar()`; และไปยังบรรทัดใหม่ เมื่อจบบรรทัด

`addstr(str)*`

เรียกไปยัง `addch()` โดยแต่ละอักขระใน str

`attroff(attrs)*`

ปิด แอทริบิว(attributes)ที่อื่น

`attron(attrs)*`

ปิด แอทริบิว(attributes)ที่อื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<code>attrset(attrs)*</code>	ตั้งค่า แอททริบิว(attributes)ชื่อนั้นเป็นค่าปัจจุบัน
<code>beep()*</code>	เสียง "บีบ"
<code>box(win, vert, hor)</code>	สร้างกล่องซึ่งระบุขนาดได้รวมทั้งระบุได้ว่าใช้ windows ไດ
<code>cbreak()*</code>	ตั้งค่าเป็น ctrl break
<code>deich()</code>	ลบอักขระ
<code>echo()*</code>	ตั้งค่าเป็น echo
<code>endwin()*</code>	จบ window
<code>getch()</code>	รับอักขระจาก tty
<code>getstr(str)</code>	รับสายอักขระจาก stdscr
<code>gexy(win, y, x)</code>	ตั้ง พิกัด (y,x) ((y,x) co-ordinater
<code>inch()</code>	ตั้งอักขระ ณ ตำแหน่งพิกัดปัจจุบัน
<code>initscr()*</code>	หน้าจอเริ่มแรก
<code>keypad(win, bf)</code>	ให้อ่านจในการใช้คีย์แพทรับข้อมูลเข้า (keypad)
<code>move(y, x)*</code>	เคลื่อน(move) ไป (y, x) บน stdscr
<code>noecho()*</code>	ไม่ตั้งค่าเป็น เอคโค(echo)
<code>nonl()*</code>	ไม่ต้องค่าเป็นบรรทัดใหม่(new - line)
<code>printw(fmt, arg1, arg2, ...)</code>	printf() ที่ไ้บน stdscr
<code>refresh()*</code>	ทำเป็นหน้าจอที่ถูกเปลี่ยนแปลงปัจจุบันใน stdscr
<code>resetty()*</code>	ทำให้แฟล็ก(flags)ของttyจากค่าที่ถูกเก็บไว้
<code>savetty()*</code>	เก็บค่า tty ปัจจุบัน
<code>scanw(fmt, arg1, arg2, ...)</code>	scanf() ที่ไ้บน stdscr

แอททริบิว(Attributes)

แอททริบิวต่อไปนี้จะถูกส่งได้โดยใช้ฟังก์ชัน

`attron()`, `attroff()`, and `attrset()`.

A_STANDOUT

การแสดงผลแต่ละเทอร์มินอล(Terminal)ทำได้ดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A_UNDERLINE	มีเส้นใต้
A_REVERSE	แสดงผลแบบตรงข้าม
A_BLINK	กะพริบ
A_DIM	ลดสว่างลงกึ่งหนึ่ง
A_BOLD	สว่างเป็นพิเศษหรือตัวหนา
A_BLANK	ไม่มีการแสดงผล
A_ALTCHARSET	อักขระอื่นๆ

แอทริบิวแบบ NLS (NLS Attributes)

แอทริบิวแบบ NLS จะถูกส่งกลับ(return)โดย inch0

A_FIRSTOF2	อักขระไบต์(byte)แรกของอักขระแบบ16บิต
A_SECOF2	อักขระไบต์(byte)ที่สองของอักขระแบบ16บิต

ปุ่มใช้งาน(Function Keys)

ปุ่มใช้งานควรสามารถถูกส่งกลับโดย getch ถ้าปุ่มนั้นสามารถใช้งานได้(enable)

หมายเหตุ : ถ้าไม่มีการกำหนด(define)ในเทอร์มินอล(terminal)ก็จะไม่สามารถส่งค่าที่กดกลับมาได้

ชื่อ(name)	ค่า(value)	ชื่อของปุ่ม(key name)
KEY_BREAK	0401	ปุ่มหยุด(break key)(unreliable)
KEY_DOWN	0402	ลูกขรทั้งสี่
KEY_UP	0403	
KEY_LEFT	0404	
KEY_RIGHT	0405	
KEY_HOME	0406	ปุ่มโฮม(Home key)(upward+left arrow)
KEY_BACKSPACE	0407	ปุ่มกดลบหลัง backspace (unreliable)
KEY_F0	0410	ปุ่มฟังก์ชันต่างๆซึ่งจองเนื้อที่ไว้ถึง64ปุ่ม
KEY_F(n)	(KEY_F0+(n))	สูตรหาฟังก์ชันต่างๆ
KEY_DL	0510	ลบบรรทัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

KEY_IL	0511	แทรกบรรทัด
KEY_DC	0512	ลบอักขระ
KEY_IC	0513	แทรกอักขระ
KEY_EIC	0514	ออกจากโหมด(mode)การแทรก
KEY_CLEAR	0515	ลบหน้าจอ
KEY_SF	0520	เลื่อน 1 บรรทัดไปข้างหน้า
KEY_SR	0521	เลื่อน 1 บรรทัดไปข้างหลัง(ย้อนกลับ)
KEY_NPAGE	0522	หน้าถัดไป
KEY_PPAGE	0523	หน้าที่แล้ว
KEY_ENTER	0527	เอนเทอร์(ENTER)เพื่อการส่ง
KEY_RESET	0531	เริ่มใหม่(reset)

ซิกแนล(signal)ที่รองรับการเปลี่ยนแปลงของวินโดว์

ทุกฟังก์ชันจะรองรับ SIGWINCH แอปพลิเคชันที่เชื่อมโยงกับ curses จะทำการวาดหน้าจอใหม่ทันทีที่ได้รับทราบว่ามีกการเปลี่ยนขนาดของ window ตัวแปร LINES และ COLUMNS จะถูกทำให้มีค่าปัจจุบัน และทำให้สามารถทำงานได้ถูกต้องตามขนาดที่แท้จริงของwinodw

แนะนำ MH (Mail Handling)

ระบบจัดการจดหมายข้อความ เป็นชุดโปรแกรมของ จดหมายอิเล็กทรอนิกส์ ถ้าคอมพิวเตอร์ของคุณใช้ระบบปฏิบัติการ ระบบปฏิบัติการยูนิกซ์ เท่านั้นเครื่องคอมพิวเตอร์เครื่องนั้นก็สามารถที่จะทำงานโดยชุดโปรแกรม ระบบจัดการจดหมาย ได้แล้วครับ ซึ่งการทำงานกับ ระบบจัดการจดหมาย นั้นจะทำได้อย่างมีประสิทธิภาพสูง, ความยืดหยุ่นในการทำงานมาก และทำงานต่างๆ ได้อย่างอิสระโดยไม่ขึ้นกับการทำงานร่วมกับโปรแกรม หรือ คลังที่รวบรวมโปรแกรมอื่นๆ ภายนอกแต่อย่างใด ทั้งนี้ยังสามารถเรียนรู้ได้ง่ายอีกด้วย

ทำไมจึงเลือกใช้ MH

เหตุผลที่ผมเลือกใช้ MH ในโครงการนี้ก็คือ ระบบจัดการจดหมาย มีความแตกต่างจาก โปรแกรมอื่นๆ และผมใช้โปรแกรมภาษาซี มาควบคุมการทำงานได้เช่นเดียวกับการควบคุมการทำงานของคำสั่งที่เรียกได้จาก ยูนิกซ์ เซลล์ หรือม เนื่องจากคำสั่งของ ระบบจัดการจดหมาย ทุกคำสั่งจะเป็นโปรแกรม แยกจากกันเดี่ยวๆ และ เซลล์ ถูกใช้เป็นเช่น ตัวแปลภาษา จึงส่งผลทำให้คุณสมบัติต่างๆ ที่ใช้กับ ยูนิกซ์ เซลล์ สามารถทำงานร่วมกับ ระบบจัดการจดหมาย ได้จึงทำให้ไม่ต้องเรียนรู้เกี่ยวกับการต่อเชื่อม เช่นโปรแกรมจัดการ จดหมาย อื่นๆ ส่วนโปรแกรมมีตัวแปลภาษาของตนเองสำหรับคำสั่งเกี่ยวกับจดหมายข้อความที่ใช้กับโปรแกรมของตน

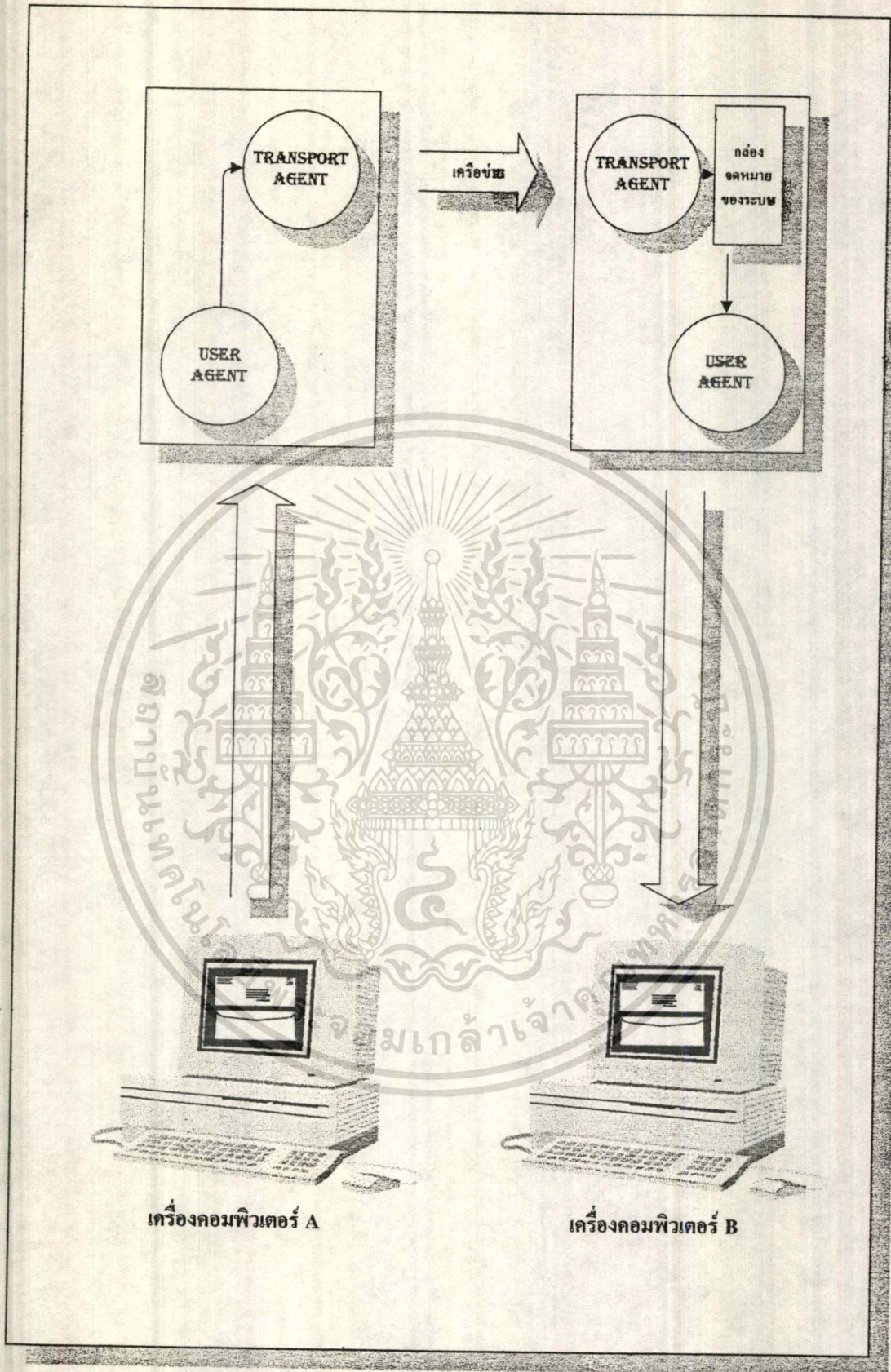
จดหมายอิเล็กทรอนิกส์คืออะไร

จดหมายอิเล็กทรอนิกส์ ซึ่งจะถูกเรียกสั้นๆ ว่า “อี-เมล” นั้นเป็นหนทางในการสื่อสารกับบุคคลต่างๆ ไม่ว่าจะใกล้ไกลเช่นไรก็ตาม เพียงแค่คุณเขียนข้อความที่ต้องการลงไป และให้ที่อยู่กับมันเท่านั้นก็จะสามารถส่งข้อความนั้นไปให้กับบุคคลที่คุณกำหนดได้แล้ว โดยที่ผู้รับ จดหมาย ไม่จำเป็นต้องใช้ จดหมายระบบ เช่นเดียวกับผู้ส่ง และผู้รับก็ไม่จำเป็นต้องใช้ระบบปฏิบัติการ ยูนิกซ์ เช่นเดียวกับผู้ส่งอีกด้วย

จดหมายอิเล็กทรอนิกส์ในระบบยูนิกซ์ ทำงานอย่างไร

โดยทั่วๆ ไปแล้ว ระบบ ยูนิกซ์ จะมีโปรแกรมที่จะจัดการกับ จดหมาย อยู่สองชนิด คือ :

1. *user agent* เป็นโปรแกรมที่ใช้โดยมนุษย์ ซึ่งจะเก็บรวบรวมข้อความใน จดหมาย ต่างๆ ที่คุณได้ส่งไป และข้อความต่างๆ ที่คุณได้รับเข้ามา
2. *transport agent* เป็น ระบบ ที่ไม่ได้ปฏิบัติงานโดยตรงโดยคำสั่งจากการคีย์ input เข้าโดยตรง มันจะรับข้อความจาก *user agent* และเส้นทางการเดินทางของข้อความเหล่านั้นไปยังปลายทางของข้อความเหล่านั้น *transport agent* โดยทั่วๆ ไป จะส่งจดหมายไปยังกล่องรับจดหมายของผู้รับผู้นั้น (ระบบกล่องรับจดหมาย คือ แท้ที่ user agent สามารถรับข้อความใหม่ๆ ซึ่งมีผู้ส่งมาให้)



ภาพแสดงเส้นทางการส่งข้อความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขอให้ท่านดูรูปของการส่งข้อความระหว่างเครื่องคอมพิวเตอร์ A และเครื่องคอมพิวเตอร์ B ซึ่งจะแสดงให้เห็นถึงสิ่งต่างๆ ที่เกิดขึ้นเมื่อทำการส่งข้อความ โดย user agent ของท่านจะรวบรวมข้อความและส่งข้อความนั้นต่อไปยัง transport agent และ transport agent ของท่านทำการจัดการส่งข้อความนั้นต่อไปยังผู้ที่ท่านต้องการจะส่งไปถึง และถ้าผู้รับข้อความนั้นไม่อยู่ใน เครือข่ายเดียวกับท่านแล้วจะทำการส่งข้อความของท่านข้าม เครือข่าย ไปยัง เครือข่ายของผู้รับนั้นด้วย จากนั้นผู้ที่ท่านส่งข้อความไปให้ท่านจะทำการเรียก user agent มาปฏิบัติงาน ผังของเขาแล้ว user agent จะทำหน้าที่นำเอาข้อความใน กล่องรับจดหมายของระบบขึ้นมาอ่าน, พิมพ์, ตอบ หรือ ลบ ข้อความเหล่านั้น

Transport Agents ของจดหมายอิเล็กทรอนิกส์

Mail transport agents (MTAs) เป็นโปรแกรมที่ทำงานเกี่ยวกับฟังก์ชันต่างๆ ที่ใช้ในงานการจัดส่ง และรับข้อความ เช่น ทำการจัดการกับ ที่อยู่ที่ระบุไว้ในจดหมายข้อความ, ควบคุมดูแลข้อผิดพลาด ต่างๆ, จัดการกับการติดต่อข้ามเครือข่ายรวมทั้งภายในเครือข่ายด้วย และหน้าที่การทำงานอื่นอีก MTAs (Mail transport agent) ที่รู้จักกันทั่วไป ได้แก่ sendmail และ MMDF เป็นต้น

User Agents ของจดหมายอิเล็กทรอนิกส์

ถ้าคุณทำการส่งและรับ จดหมายอิเล็กทรอนิกส์ เพียงจำนวนน้อยๆ คุณก็ไม่จำเป็นต้องเรียนรู้เกี่ยวกับ user agents แต่การเรียนรู้เกี่ยวกับ user agent จะทำให้ทำงานกับ ระบบจัดการจดหมาย และ xmh ได้สะดวกมากขึ้นสำหรับแม้แค่ผู้เริ่มต้นทำการเรียนรู้ก็ตาม เนื่องจากมีการทำงานร่วมกับผู้ใช้ ที่ทำให้ใช้ได้อย่างสะดวกอยู่แล้ว เมื่อคุณเริ่มที่จะใช้งาน จดหมายอิเล็กทรอนิกส์ มากขึ้นแล้วคุณควรเรียนรู้เกี่ยวกับ user agent เพื่อทำงานเหล่านั้นได้อย่างสะดวกและรวดเร็วขึ้นเป็นอันมากซึ่งจะทำให้คุณจัดการกับงานทางด้าน จดหมายอิเล็กทรอนิกส์ ได้ด้วยรูปแบบที่คุณต้องการ และทำการกำหนดรูปแบบการทำงานต่างๆ ได้อย่างที่อยากให้เป็นอีกด้วย จุดเด่นก็คือ ความยืดหยุ่นของการทำงานที่คุณต้องทำอยู่เป็นประจำนั่นเอง

ข้อเสนอแนะในการเขียนโปรแกรมเชลล์ทำงานร่วมกับ MH

สิ่งที่สำคัญที่สุดอย่างหนึ่งของการทำงานของ ระบบจัดการจดหมาย คือโปรแกรมระบบจัดการจดหมายนั้นสามารถจะทำการโปรแกรมได้อย่างอิสระ ซึ่งสามารถปฏิบัติได้จาก เชลล์ พร้อม ได้เลยเพราะ ระบบจัดการจดหมาย ไม่ใช่ monolithic (คือหนึ่งโปรแกรมรับผิดชอบทำทุกๆ อย่างที่อยู่ในขอบเขตที่รับผิดชอบ) โดย mail agent จะใช้ ระบบจัดการจดหมาย ใน เชลล์ โปรแกรม ได้ง่าย ถ้าคุณไม่ชอบ หรือสะดวกที่จะใช้ ระบบจัดการจดหมายการต่อเชื่อมคุณสามารถเขียนงานทางด้าน การต่อเชื่อมได้ด้วยตัวคุณเอง โดยการทำการเขียนโปรแกรมเข้าไปทำงานร่วมกับระบบจัดการจดหมายเพื่อให้ได้การเชื่อมต่อที่คุณพอใจ

จดหมายข้อความ

จดหมายข้อความแต่ละฉบับสามารถส่งถึงผู้รับได้พร้อมกับหลายคน โดยที่ผู้รับแต่ละคนจะได้รับสำเนาของจดหมายข้อความฉบับที่ตนได้รับมานั้น กระผมขอให้ท่านลองส่งจดหมายลักษณะหนึ่งถึงเพื่อนของท่านและตัวท่านเอง บุคคลที่ท่านระบุให้เป็นผู้รับเหล่านั้นจะได้รับสำเนาของจดหมายข้อความนั้น โดยในการส่งจดหมายข้อความดังกล่าวจะส่งไปให้เฉพาะกับโฮสต์ที่มีผู้ใช้ตามที่ระบุในจดหมายนั้นอยู่นั้น และการส่งจดหมายข้อความเหล่านี้ก็จะส่งไปในเส้นทางที่ต่างกันไปอีกด้วย

ในการส่งจดหมายข้อความผ่านทาง เครือข่ายที่เป็น ทีซีพี/ไอพี เช่น อีเทอร์เน็ต sendmail สามารถทำการส่งต่อจดหมายเหล่านั้นได้ด้วยตัวเอง จากตัวอย่างที่ท่านลองส่งจดหมายถึงเพื่อนของท่านและตัวท่านเอง เมื่อสักครู่นี้กระผมขอนำมาเปรียบเทียบกับวิธีการส่งจดหมายจริงๆ ที่ใช้กันในชีวิตประจำวัน คือสำเนาฉบับหนึ่งสำหรับตนเองและสำเนาอีกฉบับหนึ่งสำหรับเพื่อนของคุณที่อยู่ในต่างประเทศ

To : friend1, friend2@remote

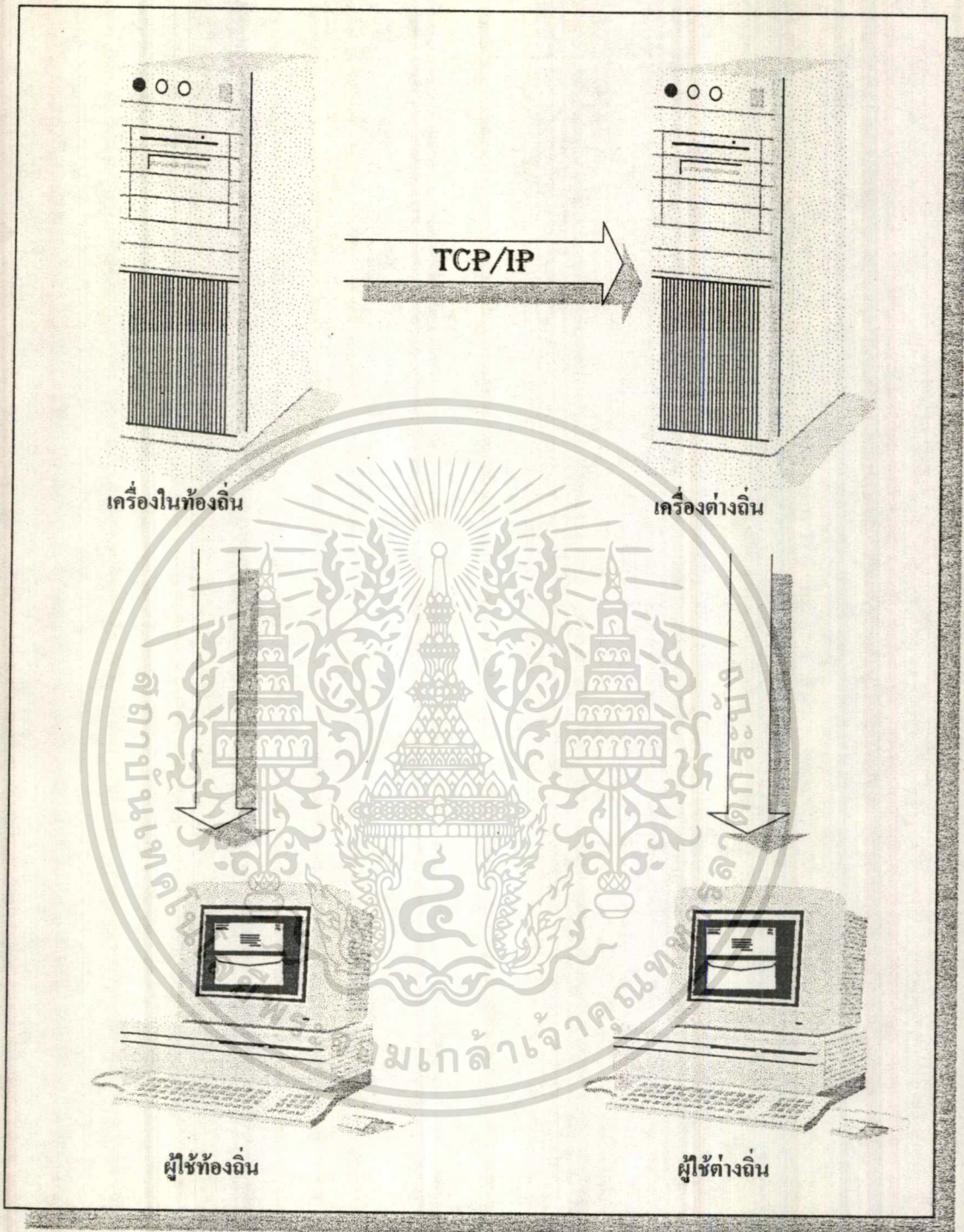
ขั้นตอนการส่งจดหมายข้อความเหล่านี้ เริ่มจากการสร้างจดหมายข้อความเหล่านี้ จากนั้นก็ทำสำเนาของจดหมายข้อความที่จะส่งนั้น จากนั้นก็จำหน้าระบุชื่อและที่อยู่ของบุคคลที่เราต้องการจะส่งไปให้ แล้วนำไปใส่ไว้ในตู้รับจดหมายเพื่อทำการส่งต่อไป โดยที่การส่งหาตัวคุณเองหรือเพื่อนที่ทำงานในสำนักงานเดียวกันกับคุณ การจัดส่งจดหมายข้อความชนิดนี้จะใช้โปรแกรม /bin/mail เป็นตัวจัดส่งให้ แต่สำหรับการส่งจดหมายที่ต้องให้บุรุษไปรษณีย์ช่วยจัดการในการส่งให้ นั้น จะใช้โปรแกรม sendmail ในการส่งจดหมายข้อความเหล่านั้นไปยังเพื่อนของท่านที่อยู่ในต่างถิ่น

ขอให้พิจารณาตัวอย่างต่อไปนี้ ถ้าท่านจะส่งจดหมายข้อความฉบับหนึ่งถึงเพื่อนของท่านสองคน โดยเพื่อนของท่านคนหนึ่งอยู่ในสำนักงานเดียวกันกับท่าน ส่วนเพื่อนอีกคนหนึ่งเป็นเพื่อนที่อยู่ในสำนักงานสาขาที่อยู่ต่างประเทศ โดยในจดหมายข้อความนั้นจะมีการจำที่หัวจดหมายดังต่อไปนี้ :

To: friend1, friend2@remote

ในกรณีนี้ส่งจดหมายข้อความให้เพื่อนของท่านที่ทำงานในสำนักงานเดียวกันกับท่านนั้นโปรแกรม sendmail จะทำการเรียกให้โปรแกรม /bin/mail มาปฏิบัติการส่งตามรูปแบบดังต่อไปนี้

/bin/mail -d friend1



การแยกจดหมายระหว่างเครื่องในท้องถิ่นและเครื่องที่อยู่ต่างถิ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ -d กำหนดว่าการส่งจดหมายนี้ จะส่งให้เฉพาะผู้รับจดหมายภายในท้องถิ่นเท่านั้น โดยจะไปกำหนดให้โปรแกรม /bin/mail ไปเพิ่มจดหมายข้อความฉบับที่ท่านส่งถึงเพื่อน เข้าไปยังผู้รับจดหมายของผู้ใช้ระบบที่ชื่อว่า friend1 (ชื่อเดียวกันกับที่ระบุไว้ให้โปรแกรม /bin/mail ขณะทำการส่ง)

ข้อสังเกต

การกำหนดที่หัวจดหมายข้อความที่ท่านระบุไว้ว่า To: friend1, friend2@remote สำหรับจดหมายที่ส่งถึงเพื่อนท่านในสำนักงานเดียวกันนั้นจะถูกส่งโดยโปรแกรม /bin/mail นั้นจะมีกำหนดที่หัวจดหมายขณะส่งเป็น -d ดังที่กล่าวมาแล้วในตอนต้นว่าเป็นการแสดงถึงในมีการส่งไปยังผู้จดหมายเฉพาะภายในท้องถิ่นเท่านั้น

จากนั้นขอให้ท่านพิจารณาการส่งจดหมายข้อความไปยังเพื่อนของท่านที่อยู่ในสำนักงานสาขาในต่างประเทศ ในระบบนี้ถือว่าสำนักงานสาขาที่อยู่ ณ ต่างประเทศนั้นอยู่ไกลจากสำนักงานของท่านมากจึงถือว่าให้ทั้งสองแห่งใช้เครื่องต่างกันซึ่งทั้งสองเครื่องนั้นอยู่ใกล้กันมากเรียกเครื่องในสำนักงานสาขาในต่างประเทศนั้นว่า เครื่องคอมพิวเตอร์ต่างถิ่น ส่วนเครื่องคอมพิวเตอร์ในสำนักงานที่ท่านอยู่เราจะเรียกว่า เครื่องคอมพิวเตอร์ท้องถิ่น ในการส่งจดหมายข้อความภายนอกท้องถิ่นนี้โปรแกรม sendmail จะเป็นผู้ส่งด้วยตัวเองโดยไม่ไปเรียก โปรแกรม /bin/mail มาทำการปฏิบัติการแทนโดยจะส่งไปยังโฮสต์ที่ระบุในส่วนของการกำหนดที่หัวจดหมายว่า @remote และจะไปเพิ่มจดหมายข้อความฉบับที่ท่านส่งไปไว้ในกล่องใส่จดหมายของระบบเครื่องคอมพิวเตอร์ต่างถิ่นนั้นโดยทำสำเนาของจดหมายข้อความที่ท่านส่งไปให้ไว้หนึ่งชุด

กระผมจะขอสรุปการทำงานของ sendmail ในการส่งจดหมายข้อความผ่านระบบเครือข่าย การทำงานของ sendmail นี้จะเหมือนกับการส่งจดหมายจริงๆ ทั่วไปในชีวิตประจำวันที่ท่านเห็นและใช้กันอยู่ ขอให้ท่านดูรูปที่แสดงถึง การสนทนาระหว่างเครื่องคอมพิวเตอร์ภายในท้องถิ่น และเครื่องคอมพิวเตอร์ต่างถิ่น

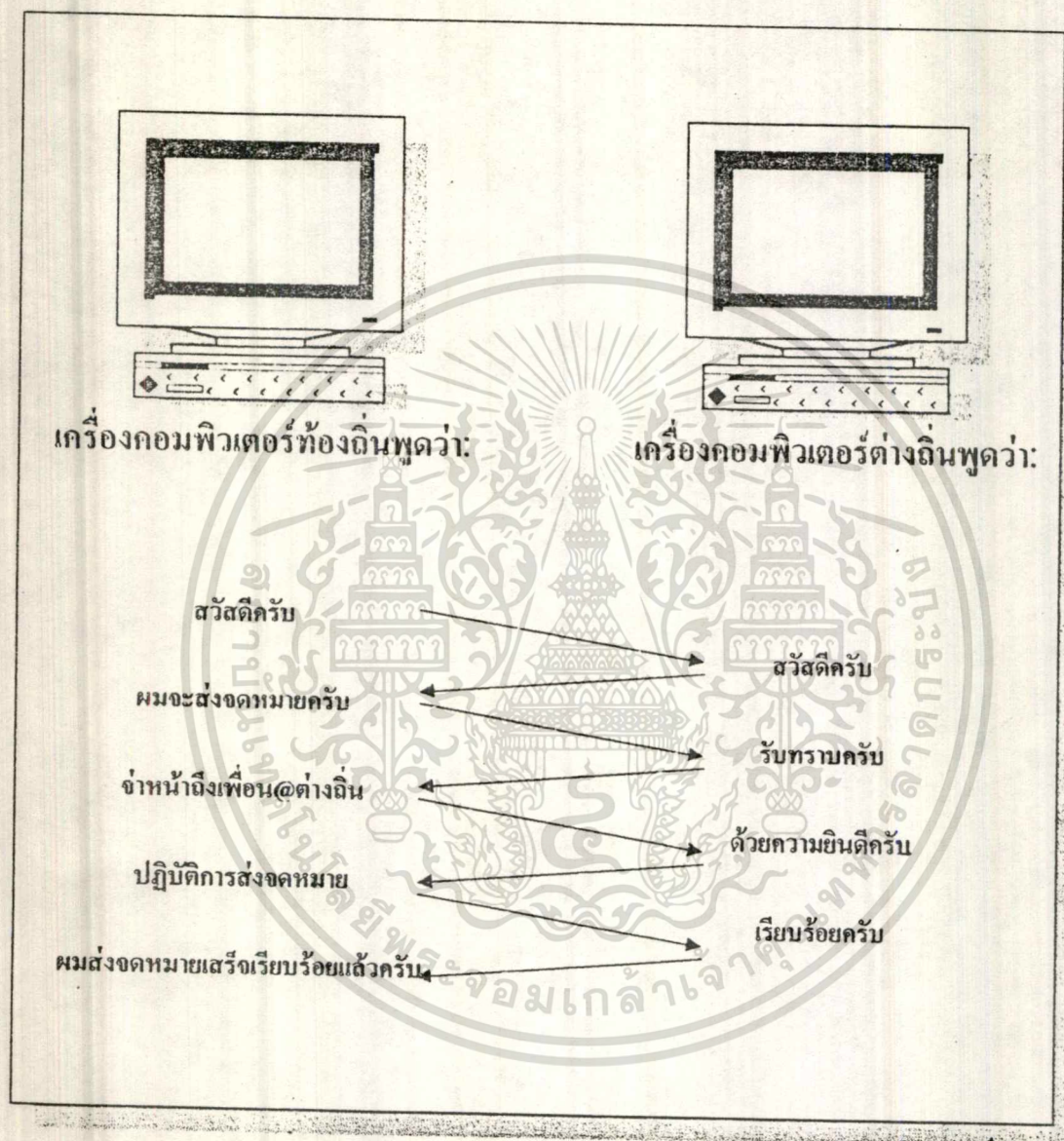
ในรูป การสนทนาระหว่างเครื่องคอมพิวเตอร์ภายในท้องถิ่น และเครื่องคอมพิวเตอร์ต่างถิ่น แสดงถึงการส่งที่เป็นการส่งผ่านระบบเครือข่าย หมายถึงส่งไปยังเครื่องคอมพิวเตอร์ที่ไม่ได้อยู่ในท้องถิ่นเดียวกันนั่นเอง ในตัวอย่างจะหมายถึงส่วนที่เป็น

To: friend2@remote

ในขั้นแรกของการส่งจดหมายข้อความไปยังเครื่องคอมพิวเตอร์ต่างถิ่นนั้นโปรแกรม sendmail จะทำการตรวจสอบที่อยู่ของผู้รับก่อนเป็นอันดับแรกว่ามีผู้รับที่มีที่อยู่ในเครื่องคอมพิวเตอร์ต่างถิ่นบ้างหรือไม่ ถ้ามีก็จะนำที่อยู่ของผู้รับเหล่านั้นจำนวนมาเพื่อทำการส่งสำเนาของจดหมายข้อความไปตามที่อยู่ที่อยู่ต่อไป โดยจะตัดเอาที่อยู่ของผู้รับที่อยู่ภายในเครื่องคอมพิวเตอร์ท้องถิ่นออกเสีย จำนวนที่เหลือก็จะเป็น To: friend2@remote ส่วนผู้รับในท้องถิ่น

คือ friend1 จะถูกตัดออกไปแล้วให้โปรแกรม /bin/mail ทำการส่งให้แทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ตัวอย่างการสนทนาระหว่างเครื่องท้องถิ่นและเครื่องต่างถิ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

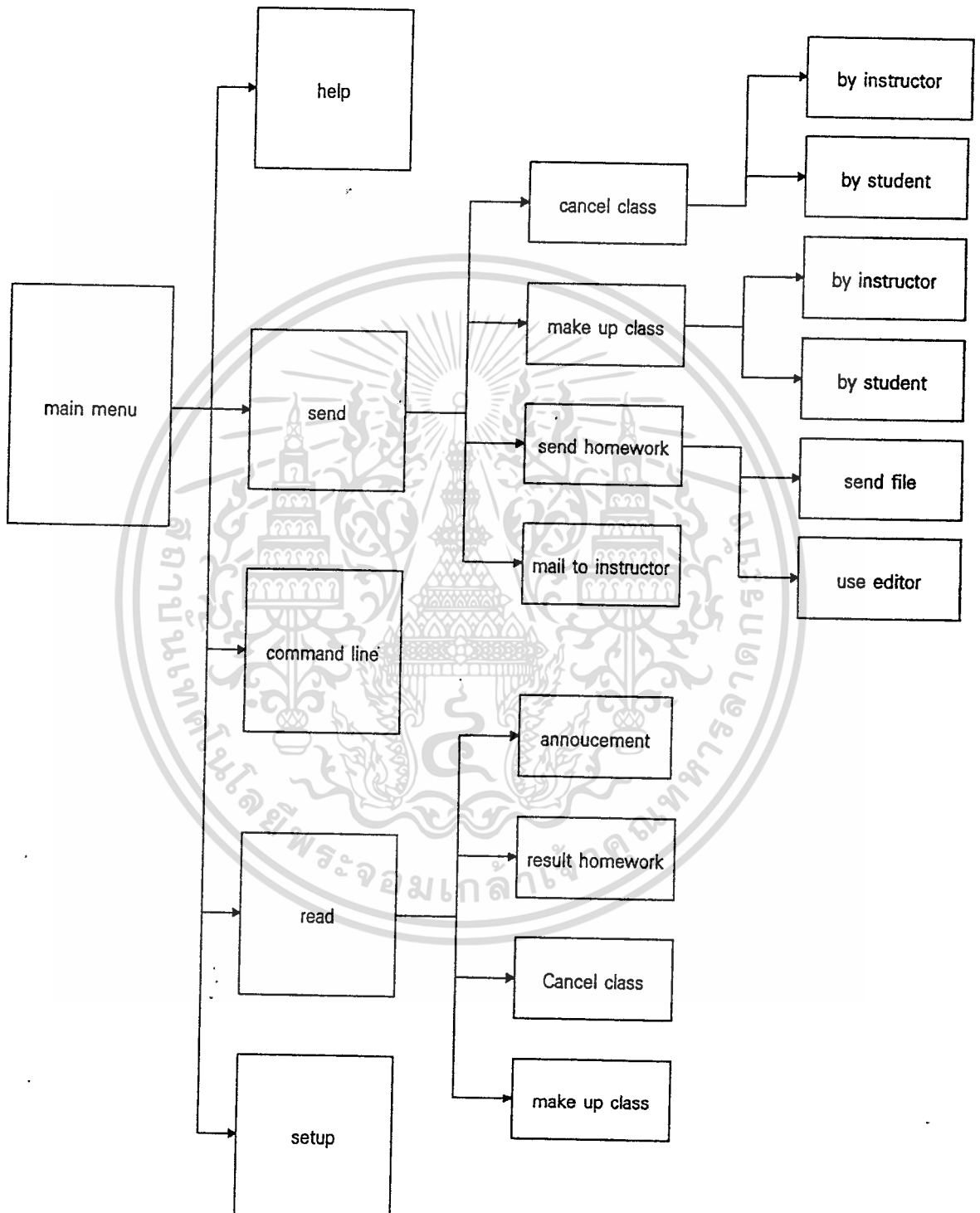
การออกแบบและการสร้าง

ในบทนี้กล่าวถึงการประยุกต์ใช้ของการเปลี่ยนแปลงทิศทางข้อมูลเข้า/ออก การกำหนดสภาวะแวดล้อมของระบบให้เหมาะสมกับโปรแกรม แสดงการบริหารจดหมาย(mail) และแสดงไฟล์ชาร์ตของการติดต่อ(interface)กับโปรแกรม รวมทั้งการโปรแกรมส่วนโปรแกรมย่อยต่างๆ



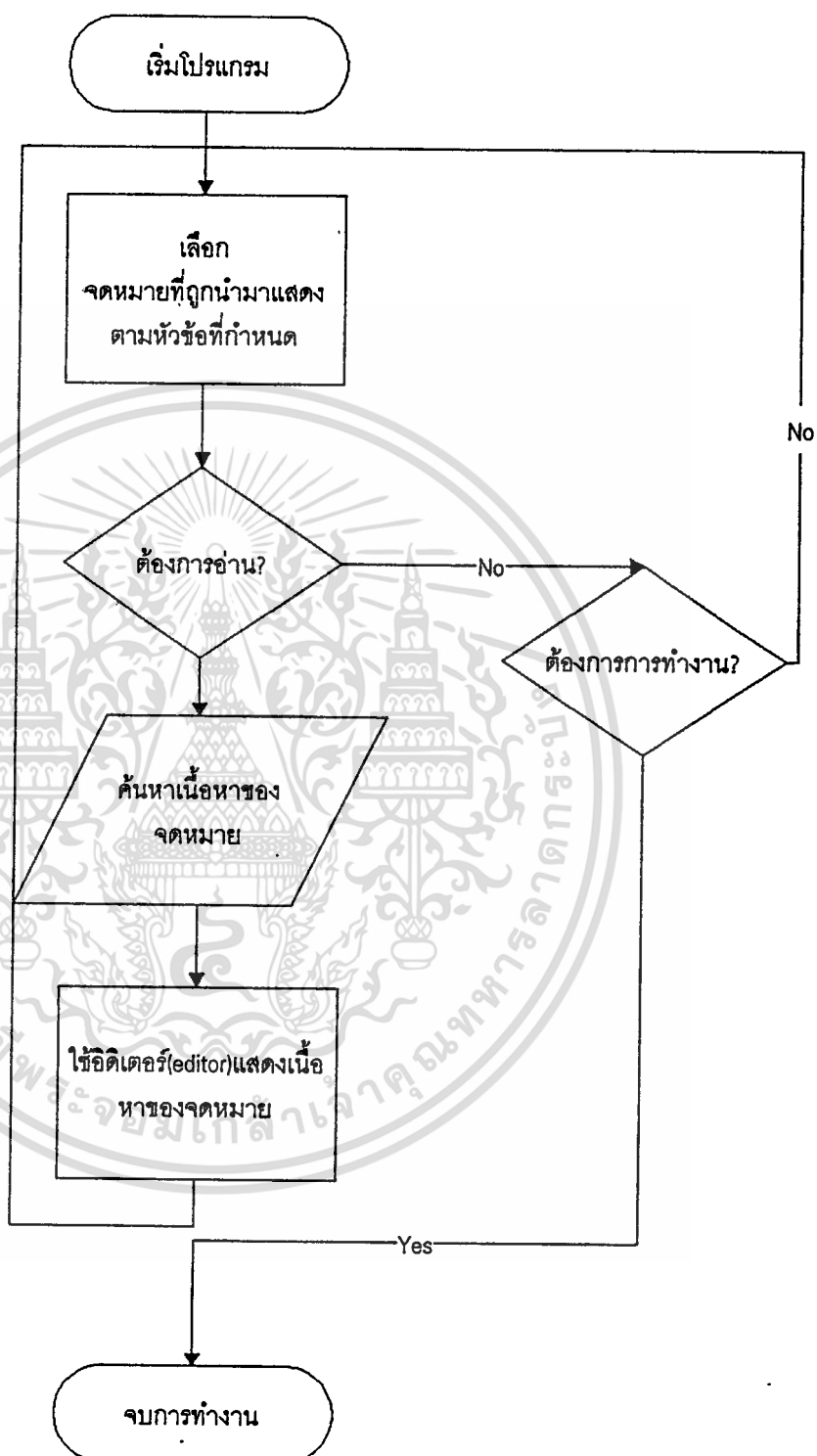
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพแสดงระบบเมนูของโปรแกรม

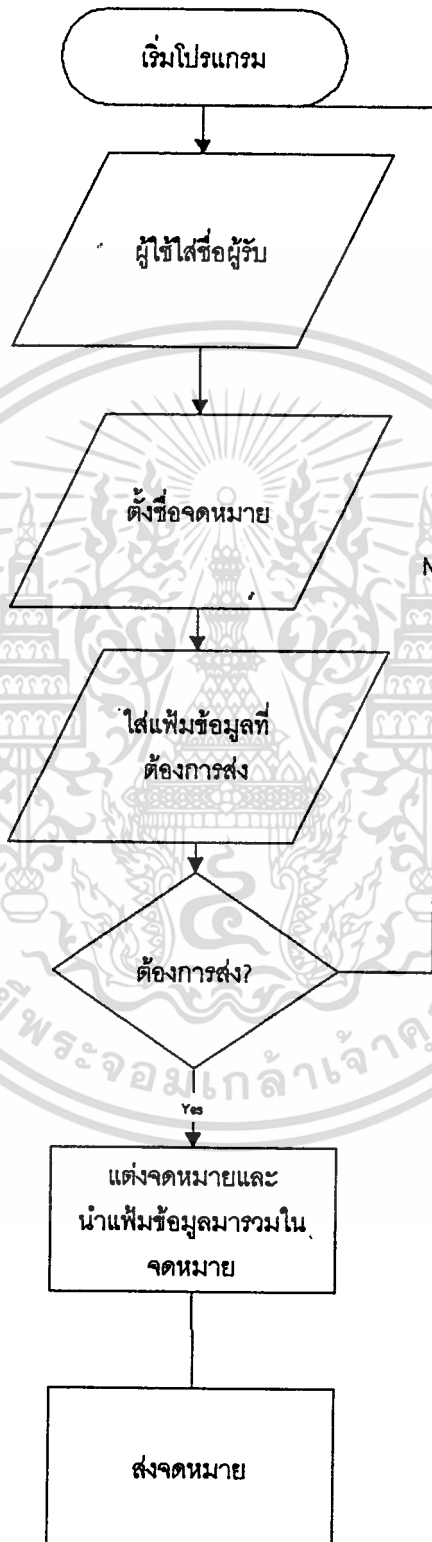


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โฟลชาร์ท(flowchart)แสดงการอ่านจดหมายตามหัวข้อ

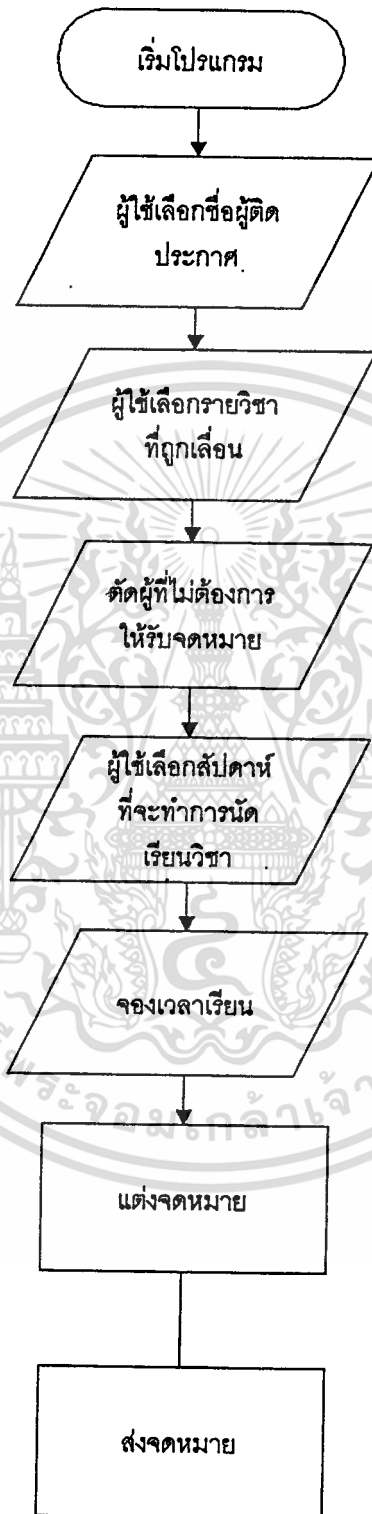


โฟลชาร์ท(folwchart)แสดงการส่งแฟ้มข้อมูลการบ้าน



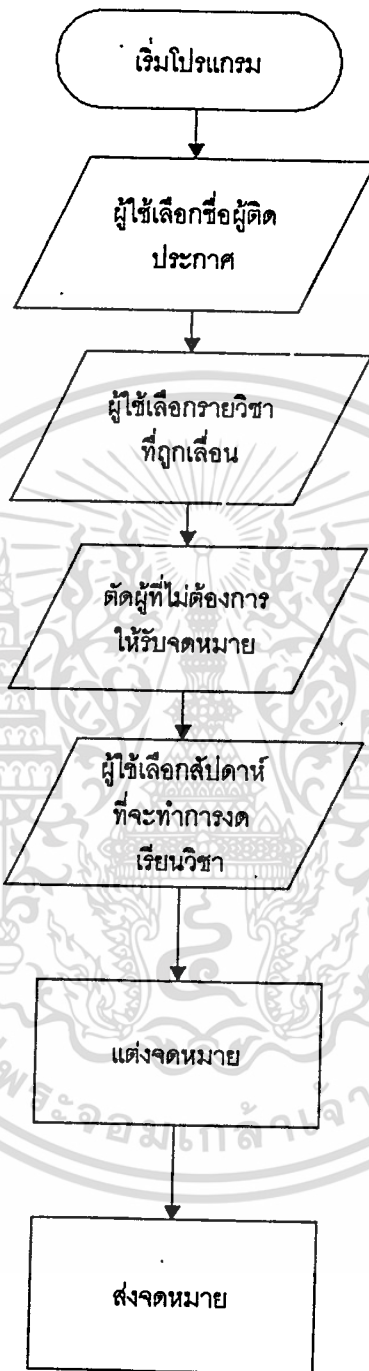
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โฟลชาร์ท(flowchart)แสดงการแต่งตั้งจุดหมายการนัดเวลาเรียนที่มีการเลื่อน



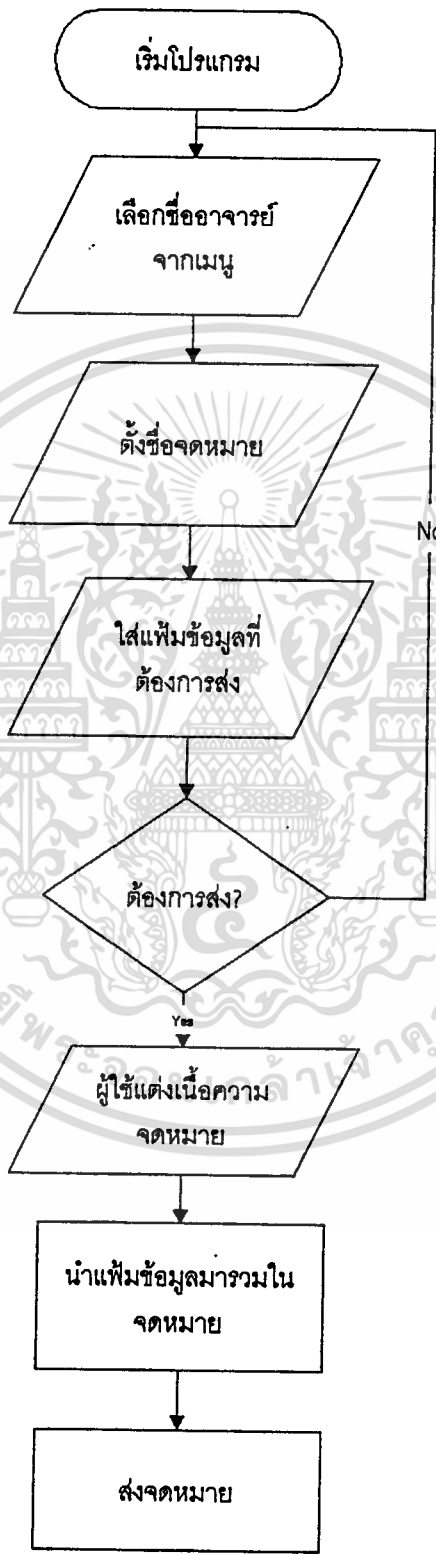
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โฟลชาร์ท(folwchart)แสดงการแต่งตั้งจดหมายการงดเรียน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โฟลชาร์ท(folwchart)แสดงการส่งจดหมายถึงอาจารย์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมย่อยที่เป็นส่วนประกอบที่ใช้ร่วมกัน

โปรแกรมที่ทำหน้าที่ตามหัวข้อประกอบด้วยส่วนโปรแกรมต่างๆที่จะอธิบายไปตามลำดับดังนี้

การตั้งsttyเพื่อให้เหมาะสมกับการใช้โปรแกรม

ก่อนใช้งานจากโปรแกรม

```
stty sane
stty intr '^O' -ixon
```

หลังใช้งาน

```
stty sane
```

ส่วนการตั้งค่าคงที่คอนโทรล(control define)

เพื่อความสะดวกในการอ้างไม่คอนโทรลในการเขียนโปรแกรม

```
#define CTRL_A 1
#define CTRL_B 2
#define CTRL_C 3
#define CTRL_D 4
#define CTRL_E 5
#define CTRL_F 6
#define CTRL_G 7
#define CTRL_H 8 /* as back_space */
#define CTRL_I 9
#define CTRL_J 10
#define CTRL_K 11
#define CTRL_L 12
#define CTRL_M 13
#define CTRL_N 14
#define CTRL_O 15 /* standard write out in our program */
#define CTRL_P 16
#define CTRL_Q 17
#define CTRL_R 18
#define CTRL_S 19
```

```
#define CTRL_T 20
#define CTRL_U 21
#define CTRL_V 22
#define CTRL_W 23
#define CTRL_X 24
#define CTRL_Y 25
#define CTRL_Z 26
```

ส่วนการยืนยันการกระทำ

เช่นถ้าต้องการออกจะถามว่า "Do you want to exit ? (y,n)." โดยปุ่มที่ผู้ใช้กดจะมีผลต่อการกระทำของขั้นตอนต่อไป

```
/* confirm :function
   return :state to go
   globol :checkexitmenu1 (declare in menu1.h)
*/
confirm(msg)
char msg[80];
(
  char c;
  int i;

  nonl();noecho();cbreak();
  attron(A_REVERSE);
  move(21,0);
  for(i=0;i<COLS;i++)
    addch(' ');
  refresh();
  move(21,0);
  addstr(msg);
  attroff(A_REVERSE);
  addch(' ');
  attron(A_REVERSE);
  refresh();
  c=getch();
  refresh();
```

```

        if(c=='y'||c=='Y')
        { addstr(" Yes ");
          refresh();
          return 1;
        }

        else if(c=='n'||c=='N')
        { if(oldcheckexitmenu1!=CTRL_X&&oldcheckexitmenu1!=CTRL_M)
            checkexitmenu1=oldcheckexitmenu1;
          else
            checkexitmenu1=CTRL_K;

          refresh();
          addstr(" No ");
          return 0;
        }

        else if(c!=CTRL_M)
          checkexitmenu1=c;
        else
          checkexitmenu1=oldcheckexitmenu1;
    }
}

```

การรับข้อมูลเป็นสายอักขระ

สามารถรับข้อมูลเป็นสายอักขระหรือรับข้อมูลเป็นปุ่มควบคุม

```

/*
readline : function
do       : receive input for getch
return   : type of character that you want to return
environm : curses
*/
readline(st,limitchar)
char st[80];
char limitchar;

    { char c,x,y,xbuf,ybuf;
      int i;

      int everb=0;

```

```

int counter=0;

refresh();

getyx(stdscr,ybuf,xbuf);

do{

    getyx(stdscr,y,x);

    if(x>=xbuf&&y==ybuf); /*maximum character is 40*/
    {

        c=readch10;

        if(c>=CTRL_A&&c<=CTRL_Z)
        /*WARN:do not use ctrl_m it is '\n'*/
        { st[0]=c;
          st[1]='\0';
          return 1;
        }
        if(x>=xbuf+limitchar)
            move(y,x-1);

        if(!((x==xbuf&&c<27)||{extrakey==1})))
        {
            addch(c);
        }
        else if(extrakey==1)
        {
            if((c==67||c==KEY_LEFT))
                move(y,x+1);

            if((c==68||c==KEY_RIGHT)&&x>=xbuf)
                move(y,x-1);
        }

        refresh();
    }

}while(c!='\n'&&c!='\r');

for(i=xbuf;i<x;i++)

```

```

{

    move(ybuf,i);

    c=inch();

    if(c==' '&&everb==0)
    {

        everb=0;

    }else

    { st[counter]=c;

        counter++;

        everb=1;

    }

    st[counter]=0;

    return 0;

}

```

การแสดงผลและตกแต่งหน้าจอ

ลบหน้าจอจากบรรทัดที่ beg ถึง บรรทัดที่ ends

```

#define MSG_LINE 21

/*
clearpage : procedure
do : clear page from row(beg) to row(ends)
*/

#include <string.h>
clearpage(beg,ends)
int beg;
int ends;
{ int i,o;

    attroff(A_REVERSE);

    for(i=beg;i<=ends;i++)
        for(o=0;o<COLS;o++)
        {

            move(i,o);

```

```

        addch(' ');
    }
    refresh();
}

```

แสดงข้อความแบบตรงข้ามกับสีปัจจุบัน

```

/*
line_reverse : function
return : none
do : put message in middle of screen which programmer define
*/
int line_reverse(row,topic)
int row;
int topic;
{
    int i;

    attrset(A_DIMIA_REVERSE);
    /* set color reverse if vt100 dim if hp */
    attron(A_DIMIA_REVERSE);
    /* on attr */
    move(row,0);
    /* gotoxy(programmer define,left);*/

    for(i=0;i<(COLS-strlen(topic))/2;i++)
        /* fill blank to middle */
        addch(' ');

    move(row,(COLS-strlen(topic))/2);
    /* prepare to write string in middle */
    addstr(topic);
    /* write string */
    move((COLS-strlen(topic))/2,COLS); /* go to end of string */

    for(i=(COLS-strlen(topic))/2+strlen(topic);i<COLS;i++)
        /* fill blank to end */

```

```

    addch(' ');
    refresh();
}

```

แสดงข่าวสารแบบมีสีตรงข้ามกับหน้าจอปกติ

```

/* msg_reverse : function
   return : none
   do : put message in middle of screen which programmer define
*/

```

```

int msg_reverse(row,topic)
int row;
int topic;
{
    int i;
    int x,y;
    refresh();
    getyx(stdscr,y,x);
    attrset(A_DIMIA_REVERSE);
    /* set color reverse if vt100 dim if hp */
    attroff(A_DIMIA_REVERSE);
    /* on attr */
    move(row,0);
    /* gotoxy(programmer define,left);*/

```

```

    for(i=0;i<(COLS-strlen(topic))/2;i++)
    /* fill blank to middle */
    addch(' ');
    attron(A_DIMIA_REVERSE);
    move(row,(COLS-strlen(topic))/2);
    /* prepare to write string in middle */
    addstr(topic);
    /* write string */
    attroff(A_DIMIA_REVERSE);
    move((COLS-strlen(topic))/2,COLS);

```

```

/* go to end of string */

for(i=(COLS-strlen(topic))/2+strlen(topic);i<COLS;i++)
/* fill blank to

                                end */

addch(' ');
move(y,x);
refresh();
}

```

ระบบการช่วยเหลือ(help)แบบต่างๆ

ระบบช่วยเหลือที่ใช้แบ่งได้ดังนี้

- 1). ระบบช่วยเหลือแสดงปุ่มควบคุม
- 2). ระบบช่วยเหลือระหว่างขั้นตอนการทำงาน
- 3). ระบบช่วยเหลือของตัวโปรแกรม

ระบบช่วยเหลือปุ่มควบคุม

จะแสดงปุ่มควบคุมต่างๆว่ามีหน้าที่อย่างไรในขั้นตอนการทำงานนั้นโดยมีลักษณะเป็นสแตก(stack)คือสามารถนำข้อมูลเข้าออกได้โดยถ้าเข้าก่อนจะได้อยู่ล่างสุดเข้าทีหลังจะอยู่บนสุดระบบนี้ทำงานโดยใช้ภาวะแวดล้อมแบบวินโดว์

```

chelp(keydel,key,help)
int keydel;
char key;
char help[10];
{
    int i,o;
    int bufmax;
    int max;
    int counter;

    static char ckey[13]={' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '};
    static char chelp[12][15];

    for(i=0;i<=12;i++)
        if(ckey[i]==key)

```

```

        ckey[i]=' ';

    if(keydel!=1)
    {
        i=0;
        while(ckey[i]!=' '&i<12)
            i++;
        ckey[i]=key;
        strcpy(chelp[i],help);

        if(i>=12)
            Error("Your menu overflow");
    }
    for(i=0;i<12;i++)
    {
        while(ckey[i]!=' ')
            i++;
        for(counter=i;counter<11;counter++)
        {
            ckey[counter]=ckey[counter+1];
            strcpy(chelp[counter],chelp[counter+1]);
        }
    }

    clearpage(22,23);

    max=0;
    while(ckey[max]!=' ')
        max++;
    bufmax=6;

    if(keydel!=3)
        for(i=0;i<max;i++)

```

```

{
    if(ckey[i]!=' ')
    {
        if(i%2==0)
        {
            o=(i/2)*(COLS/bufmax);
            move(22,o);
            attron(A_REVERSE);
            addch('^');
            addch(ckey[i]);
            attroff(A_REVERSE);
            addch(' ');
            addstr(chelp[i]);
        }
        else
        {
            o=((i-1)/2)*(COLS/bufmax);
            move(23,o);
            attron(A_REVERSE);
            addch('^');
            addch(ckey[i]);
            attroff(A_REVERSE);
            addch(' ');
            addstr(chelp[i]);
        }
    }
}

refresh();

/* manage by compress line*/
}

```

ระบบช่วยเหลือระหว่างขั้นตอนการทำงาน

ใช้การแสดงอักษรข้อความสื่อให้เห็นว่าผู้ใช้ควรทำอะไรจึงทำงานได้ตามวัตถุประสงค์ ซึ่งโดยปกติใช้การส่งข้อความไป

```
#define MSG_LINE 21

main()

#include "msg_reverse.h"

{
msg_reverse(MSG_LINE," msg ");
}
```

ระบบช่วยเหลือของโปรแกรม

เมื่อกดปุ่ม CTRL-H โปรแกรมย่อยส่วนนี้จะเก็บหน้าจอลงในหน่วยความจำแล้วเปิดแฟ้มข้อมูลขึ้นเพื่อแสดงข้อมูลในรูปของการเลื่อนทีละหน้า และเมื่อออกจากระบบนี้ระบบจะคืนหน้าจอเดิมโดยไม่สนใจแอททริบิวต์(attribute)

```
#include <stdio.h>
/*
getscreen : procedure
effect   : globol screen[][]
do       : save current video in buffer
*/
char screen[22][80];
getscreen()
{
    int x,y;
    for(y=1;y<21;y++)
        for(x=0;x<80;x++)
        {
            move(y,x);
            screen[y][x]=inch();
        }
}
/*
putscreen : procedure
noneffect : globol screen[][]
do        : restore current video in buffer
```

```

*/
putscreen()
{
    int x,y;

    attroff(A_REVERSE);

    for(y=1;y<21;y++)
        for(x=0;x<80;x++)
        {
            move(y,x);
            addch(screen[y][x]);
        }
    refresh();
}
/*
npage : function
do : save page in memory by check not over page seting
return : if not overflow charactor count , if overflow -1
*/
npage(pnum,number_page,page)
int pnum;
int number_page;
char *page;
{
    int i=0;
    int buf;
    int checkerr=0;
    buf=number_page*pnum;
    while(i<buf&&checkerr<20000)
    {
        if(page[checkerr]==10)
            i++;
        checkerr++;
    }
}

```

```

        if(checkerr>=20000)

            return -1;

        else

            return checkerr;

    }

/*
hv   : procedure
do   : manage help menu page scroll up down and loadfile *name
environment : curses
*/
hv(name)
char *name;
{

    int i=0;
    char *page;
    FILE *fp;
    int counter;
    int x,y;
    int c;
    int buf;
    int o;
    char *piconame;

    piconame=malloc(100);
    page=malloc(20000);
    getscreen();
    fp= fopen(name,"r");
    getyx(stdscr,y,x);
    msg_reverse(22,"[Help]");
    line_reverse(23,"N':Next page 'P':Previous page 'X': eXit ");
    refresh();
    nonl();cbreak();noecho();

    c=CTRL_P;

    o=0;

```

```

move(1,0);

while(!feof(fp))
{
    i=0;
    strcpy(piconame,readf(fp));

    while(piconame[i]!='\n'&& piconame[i]!=0)
    {
        page[0]=piconame[i];
        o++;
        i++;
    }
    refresh();
}
buf=0;
move(1,0);
counter=0;
c='p';

do{
    counter=0;

    if(c=='p'&&c=='P')
    {
        buf--;
    }

    if(c==CTRL_Mllc=='N'&&c=='n')
    {
        buf++;
    }

    if(buf<0)

```

```

    {
        msg_reverse(22,"You already at top of help");
        buf=0;
    }
    i=npage(buf,20,page);

    if(i==1)
    {
        msg_reverse(22,"You already at bottom of help");
        buf=buf-1;
    }
    i=npage(buf,20,page);

    if(c==CTRL_Milc=='n' || c=='N' || c=='P' || c=='p')
    {
        clearpage(1,21);
        while(counter<20&& i<20000&& page[i]!=0)
        {
            if(page[i]=='\n')
            {
                counter++;
                i++;
                printf("\n");
            }
            else
            {
                if(page[i]!=0)
                {
                    if(page[i]!=10)
                        addch(page[i]);
                } else
                    addch(' ');
                i++;
            }
        }
    }
}

```

```

    }
    if(i>20000)
        Error("can not find back n before end of array size");
    }
    refresh();
    c=getch();
}
while(c!='X'&&cl!='x');
putscreen();
fclose(fp);

if(oldcheckexitmenu1!=CTRL_M)
    checkexitmenu1=oldcheckexitmenu1;
}

```

รับข้อมูลแบบตัวเดียวไม่รอการกดเอนเทอร์(ENTER)

บนเทอร์มินอล(terminal)ใดๆ เป็นโปรแกรมพื้นฐานในการอ้างถึงปุ่มต่างๆที่กดแล้วแสดงผลทันทีไม่ต้องมีเอนเทอร์

```

/* readch1 : function
   return : character
   do      : receive key from user
*/
/* .Warning : don't use ESCAPE key */
int readch1()
{
    int c1, counter=0;
    savetty(); standend();nonl();cbreak();noecho();keypad(stdscr,TRUE);
    extrakey=0;
    c1=getch();
    counter=counter+1; /* receive key from user */
    if(c1==27)
    {
        getch();
        c1=getch();
        extrakey=1;
    }
}

```

```

}

resetty();

return c1;

/* return toupper(c1); */ /* change to big character Upper case */
}

```

ชุดการทำงานสร้างเมนูเนกประสงค์(general propose menu)

สร้างเมนูที่ทำงานได้โดยรับสายอักขรแบบอาร์เรย์(array)มาแสดงผลในรูปแบบที่สามารถเลือกได้ว่าต้องการข้อมูลตัวใดและจะกลับโดยปกติหรือกลับโดยใช้ปุ่มควบคุม

```

#include <ctype.h>
#include <string.h>

#define MAX_MENU1 200 /* max of array on menu */
int current_item1; /* global reserve for this menu1.h */
/* extrakey : extrakey such as four arrow key
longest_1: longest of menu
*/
int extrakey;
int longest_1;
/* two globol variable use as menu quit method */
char oldcheckexitmenu1=0;
char checkexitmenu1=0;
/* select : function
return : string that selected
do : receive user input character or cross to select menu
*/
char *selects(menu, menu_length,length)
char *menu[MAX_MENU1];
int menu_length;
int length;
{
    int c=0;
    int x,y;
    int xb,yb;

    longest_1=0;

```

```

checklongest_1(menu,menu_length);
savetty(); /* save tty of system*/
cbreak();nonl();noecho();standend();
/* set use Ctrl-break and no echo to getch */
keypad(stdscr,TRUE); /* enable keypad to use extra keyboard*/
msg_reverse(21,["Press UP and DOWN key"]);
getyx(stdscr,y,x); /* from where */

do{
    move(y,x);
    /* goto where */
    current_item1=checkkey1(c,menu_length,menu,current_item1);
    /* check current key to translate mean*/
    putmenu1(menu,menu_length,length,current_item1);
    /*display menu*/
    refresh();
    c=readch1();
}while(c!='\n'&&c!='\r'&&!(c>=CTRL_A&&c<=CTRL_Z));
resetty();
/* return tty environment */
oldcheckexitmenu1=checkexitmenu1;
checkexitmenu1=c;
return menu[current_item1+1];
/* return string selected*/
}

/* checklongest_1 : procedure
do           : find the longest of set of string in menu
effect: longest_1
*/
checklongest_1(menu,length)
char *menu(MAX_MENU1);
int length;
{ char remle=0;

```

```

int i;
int counter;

for(i=1;i<length;i++)
{
    counter=0;
    while(menu[i][counter]!=0)
    {
        counter++;
    }
    if(counter>remle)
        remle=counter;
}
longest_1=remle;
refresh();
}

/* checkkey1 : function
return : mean of key for menu selecting
do : analyze key (c)
*/
int checkkey1(c,menu_length,menu,current_item1)
int c;
int menu_length;
int current_item1;
char *menu[MAX_MENU1];
{
    int i;
    if(current_item1==0&&(c==KEY_UP||c=='A'&&extrakey==1))
        /* press UP on TOP*/
        {
            current_item1=current_item1; /* stay top*/
            msg_reverse(21,"You are at the highest line!");
        }
    else if(current_item1==menu_length-2&&(c==KEY_DOWN||c=='B'&&extrakey==1)))

```

```

    puteadge(menu,'-');
    linebegin=current_item1/length;
    linebegin=linebegin*length;

    for(i=linebegin;i<length+linebegin;i++)
    {
        move(y+counter++,x);
        if(i<menu_length-1)
            putmenulines(menu,current_item1,i);
        else
            puteadge(menu,' ');
    }
    move(y+length,x);
    puteadge(menu,'-');
    move(y+current_item1%length,x);
    refresh();
}

```

```

int puteadge(menu,c)
char *menu[];
char c;
{ int i;
    attrset(A_REVERSE);
    attroff(A_REVERSE);
    i=0;
    for(i=0;i<longest_1+1;i++)
        addch(c);
    addch(c);
}

```

```

int putmenulines(menu,current_item1,i)
char *menu[];
int current_item1;
int i;
{ int o;

    int counter=0;

```

```

attrset(A_REVERSE);
attroff(A_REVERSE);

if(current_item1==i)
{
    attron(A_REVERSE);
}else
{
    attrset(A_REVERSE);
   attroff(A_REVERSE);
}
o=0;
addch(' ');
do{
    if(menu[i+1][o]!='_')
        addch(menu[i+1][o]);
    else addch(' ');
    o++;
    counter++;
}while(menu[i+1][o]!=0);

for(o=0;o<(longest_1-counter+1);o++)
    addch(' ');
}

```

ดราฟท์(draft)เพิ่มข้อมูลที่ถูกส่ง

เมื่อโปรแกรมทำงานเสร็จและมีจดหมาย(mail)ต้องส่งจำเป็นต้องสร้างเพิ่มข้อมูลชื่อดราฟท์และมีรหัสแบบดราฟท์ ซึ่งถูกวางไว้ที่ตำแหน่ง \$HOME/Mail/draft โดยมีรูปแบบดังนี้

To: name
 CC: other name
 Subject: subject name

 Subject

Makeup-class

Cancel-class

data

โปรแกรมย่อยในการส่งกราฟท์

สามารถปรับปรุงให้เข้ากับการส่งแบบต่างๆได้

```
makedraft(week,makex,tcancel,name,member)

char *week;
char *makex;
char tcancel[MAX_MENU1];
char name[35];
char *member[20];
{
    FILE *fp;
    int i;

    fp=fopen("HOME/Mail/draft","w");
    /*To :*/
    fprintf(fp,"To: %s\n",name);
    /* CC : */
    fprintf(fp,"CC:");
    for(i=1;i<=member[0][0];i++)
    {
        if(strcmp(member[i],name)!=0)
            if(member[i][0]!='\0')
            {
                fprintf(fp," %s",member[i]);
                if(i<member[0][0]-1)
                    fprintf(fp,";");
            }
    }
    fprintf(fp,"\n");
    /* Subject */
    fprintf(fp,"Subject : Makeup-class\n");
}
```

```

fprintf(fp,"_____\n");
/* Message */
i=0;

while(tcancel[i]!=0)
{
    if(tcancel[i]=='_')
        tcancel[i]=' ';
    i++;
}
i=0;
while(week[i]!=0)
{
    if(week[i]=='_')
        week[i]=' ';
    i++;
}
fprintf(fp," For Cancel %s\n",tcancel);
fprintf(fp," Make up as %s\n",makex);
fprintf(fp," in %s\n",week);
fclose(fp);
}

```

เปิดเพิ่มข้อมูลและการอ่าน

การเปิดเพิ่มข้อมูลแบบตรวจสอบและแสดงข้อผิดพลาด

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
/* fopen : function
```

```
return : file that opened
```

```
do : openfile
```

```
from : The Unix programming page 182
```

```
*/
```

```
FILE *efopen(file,mode)
char *file,*mode;
{ FILE *fp,*fopen();

    if((fp=fopen(file,mode))!=NULL)
        return fp;
    fprintf(stderr,"0001:can't open file %s %s",file,mode);
    exit(1);
}
```

นำแฟ้มข้อมูลมาอ่านลงบัฟเฟอร์(buffer)

```
#define MAX_READ_ONCE 100
char buf_readf[MAX_READ_ONCE];
/* readf : function
   return: string that read
   do : read string from file fp by pointer
   bug : 1).don't declare other *pointer it will be core dump
         2).use globol value buf_readf[MAX_STRING]
*/
char *readf(fp)
FILE *fp;
{
    fgets(buf_readf,sizeof buf_readf,fp);
    return buf_readf;
}
```

ตัวอย่างแฟ้มข้อมูลที่ใช้เป็นฐานข้อมูล

แฟ้มข้อมูลของอาจารย์ มีการใช้อักษรพิเศษในการควบคุมการอ่านโดยโปรแกรมที่ต้องการใช้

- 1 : จำนวนสมาชิกทั้งหมด
- 2 : ชื่อสมาชิก no.x
- 3 : ตารางสอนแต่ละวิชา
- 4 : จบตารางสอน โดย เครื่องหมาย!
- 5 : เริ่มตารางสอนใหม่

4

teacher1

mon_10:00-12:00_L2_teach_4d

tue_10:00-12:00_L1_teach_3d

wed_13:00-15:00_LA1_teach_3d

|

teacher2

mon_10:00-12:00_L2_teach_4d_____

mon_10:00-12:00_L2_teach_3d_____

tue_10:00-12:00_L1_teach_4d_____

wed_13:00-15:00_LA1_teach_4d_____

thu_10:00-12:30_B-A_teach_3d,4d

|

teacher3

mon_13:00-15:00_L2_teach_4d

tue_13:00-15:00_L1_teach_3d

wed_09:00-12:00_LA1_teach_3d

thu_13:00-15:00_BA_teach_4d

|

teacher4

mon_17:00-19:30_L2_teach_2p

wed_17:00-19:30_L3_teach_2p

thu_10:00-12:30_B-A_teach_3d,4d,2p

|

เพิ่มข้อมูลของนักศึกษา มีรูปแบบเดียวกับเพิ่มของอาจารย์

3

4d

mon_10:00-12:00_L2_by_teacher1,teacher2____

mon_13:00-15:00_L2_by_teacher3_____

tue_10:00-12:00_L1_by_teacher2_____

wed_13:00-15:00_LA1_by_teacher2_____

thu_10:00-12:30_B-A_by_teacher2_____

```
thu_13:00-15:00__BA__by_teacher3_____
```

```
!
```

```
3d
```

```
mon_10:00-12:00__L2__by_teacher2____
```

```
tue_10:00-12:00__L1__by_teacher1____
```

```
tue_13:00-15:00__L1__by_teacher3____
```

```
wed_09:00-12:00__LA1_by_teacher3____
```

```
wed_13:00-15:00__LA1_by_teacher1____
```

```
thu_10:00-12:30__B-A_by_teacher2____
```

```
!
```

```
2p
```

```
mon_17:00-19:30_L2__by_teacher4_____
```

```
wed_17:00-19:30_L3__by_teacher4_____
```

```
thu_10:00-12:30_B-A_by_teacher4,teacher2
```

```
!
```

```
/* don't use control character '_' if want to use please use '*'.
```

```
D
```

ออกจากโปรแกรมเพราะความผิดพลาด

เมื่อเกิดข้อผิดพลาดรุนแรง ถ้าไม่สามารถดำเนินการต่อได้ต้องออกจากโปรแกรม

```
Error(c)
```

```
char c[100];
```

```
{
```

```
msg_reverse(21,c);
```

```
refresh();
```

```
endwin();
```

```
fprintf(stderr,"0001:can't open file %s %s",file,mode);
```

```
exit(0);
```

```
}
```

การส่งแฟ้มข้อมูลผ่านทางจดหมาย(mail)

มีลักษณะติดต่อกับผู้ใช้แตกต่างกับการเลื่อนเวลาเรียนและการยกเลิกเวลาเรียนเพราะใช้ปุ่มลูกศรควบคุมการใส่ข้อมูลเพราะไม่สนใจในการดึงข้อมูลที่อ้างอิงกัน

โปรแกรมย่อยที่ใช้งานจะใช้ตามโปรแกรมที่ได้อธิบายมาแล้วในส่วนนี้จะแสดงในส่วนของการเปลี่ยนชั้นการทำงานและความสามารถในการทำงานที่เพิ่มเข้ามา

เตรียมหน้าจอให้อยู่ในลักษณะวินโดว์และปรับปรุงเทอร์มินอลให้เหมาะสมกับโปรแกรม

```
char *str[MAX_FILE];
int wantexit=0;
initscreen()
{
    system("/smod"); /*smod is change to suit mode in my program by sty*/
    initscr();
    line_reverse(0," Send file homework");
    chelp(3,'H',"Help");
    chelp(3,'X',"eXit");
    chelp(3,'O',"WriteOut");
    chelp(0,'S',"Sent mail");
}
```

ส่งไปยังผู้ใด

เป็นการอินพุตสายอักขระ

```
to_0
{ char *st;
  int i;
  int buf;

  st=malloc(80);
  move(2,2);
  attron(A_REVERSE);
  addstr(" To :");
  move(2,10);
  refresh();

  buf=readline(st,60);
```

```

move(2,2);
attroff(A_REVERSE);

addch(' ');
addch('T');
addch('o');
addch(' ');
addch(':');
refresh();

move(20,0);
printw(" ");
return buf;
}

```

หัวข้อที่ใช้ในการส่ง
เป็นการอินพุตสายอักษร

```

subject()
{ char *st;
  int buf;
  st=malloc(80);
  move(4,2);
  attron(A_REVERSE);
  addstr(" Subject :");
  move(4,16);
  refresh();
  buf=readline(st,60);
  move(4,2);
  attroff(A_REVERSE);
  addstr(" Subject :");
  move(20,0);
  printw(" ");
  refresh();
  return buf;
}

```

วิเคราะห์โปรแกรมที่ใช้กดว่ามีความหมายอย่างไร

```

ananext(i,now)
int i;
int now;
{
    if(wantexit==1)
        return 0;
    if(i==CTRL_X)
        return 0;
    else if((i=='A' || i==KEY_UP) && now>1)
        return now-1;
    else if((i=='B' || i==KEY_DOWN) && now<3)
        return now+1;
    else return now;
}

```

เพิ่มข้อมูลที่ต้องการส่ง

สามารถแทรก(insert)ลบ(delete)หรือของเพิ่มข้อมูลนั้นๆได้

```

attachmenu(length)
int length;
{ int buf;
    refresh();
    attron(A_REVERSE);
    move(6,2);
    addstr(" attachment :");
    move(6,18);
    selects(str,length,10);
    attroff(A_REVERSE);
    move(6,2);
    addstr(" attachment :");
    return checkexitmenu1;
}

insert(length)

```

```

int length;
{

int i;
char *buf;
char *buf1;
int ret;

    buf1=malloc(80);
    buf=malloc(100);
    attron(A_REVERSE);
    move(18,2);
    addstr("Insert file");
    move(18,16);
    attroff(A_REVERSE);
    addstl(" ");
    move(18,16);
    refresh();
    ret=getline(buf1,79);
    if(ret==CTRL_E)
        wantexit=2;
    if(ret==CTRL_X)
        wantexit=1;

    if(ret!=CTRL_X&&buf1[0]!=0)
    {

        buf[0]='l';
        buf[1]='s';
        buf[2]=' ';
        i=3;
        while(buf1[i-3]!=0)
        {

            if((buf1[i-3]>='A'&&buf1[i-3]<='z')||buf1[i-1]=='*')
            {

                buf[i]=buf1[i-3];

```

```

    } else
        buff[i]=' ';
    i++;
}
buff[i]=' ';
buff[i+1]='>';
buff[i+2]='t';
buff[i+3]='e';
buff[i+4]='m';
buff[i+5]='p';
buff[i+6]= 0;
move(21,30);
refresh();
msg_reverse(21,"While loading file");
if(system(buff)!=0)
{
    move(20,0);
    msg_reverse(21,"Error:file not found");
}
else
    msg_reverse(21,"Loading finished");
/* prepare insertfile*/
length=loadstr(length);
}
else
{
    msg_reverse(21,"not load file beacuse input is NULL");
    sleep(1);
}
attroff(A_REVERSE);
move(18,2);
addstr(' ');
refresh();
return length;
}
delete(length)

```

```
int length;
```

```

(
    int i;

    clearpage(6,15);

    if(length>0)
        for(i=current_item1+1;i<length;i++)
        {
            strcpy(str[i],str[i+1]);
        }

    free(str[length]);

    if(length>0)
        return length-1;
    else
        return 0;
}

loadstr(length)
int length;
{
    FILE *fp;
    int i,o=0;

    fp=fopen("temp","r");
    if(length<0)
        length=0;

    while(!feof(fp))
    {
        str[length]=malloc(60);
        strcpy(str[length],readf(fp));
        o=0;
        while(str[length][o]!=0)
        {
            if(str[length][o]<CTRL_Z)
                str[length][o]=' ';
            if(str[length][o]==91)
            {
                str[length][0]='!';
            }
        }
    }
}

```

```

        str[length][o+1]=0;
    }
    o++;
}
length++;
}
close(fp);
system("rm temp");
return length-1;
}

```

บริหารขั้นตอนการทำงานของงาน

```

main()
{
    int now=1, endby, buf,oldnow=1;
    int length=1;
    initscreen();
    do{
        refresh();
        switch(now)
        {
            case 1 : msg_reverse(21,"[Fill User ID for target]");
                    chelp(0,'H',"Help");
                    buf=to_0;
                    refresh(); break;
            case 2 : msg_reverse(21,"[Fill Subject name]");
                    chelp(0,'H',"Help");
                    buf=subject(); break;
            case 3 : msg_reverse(21,"[Use CTRL_I to insert file and CTRL_D to delete file]");
                    chelp(0,'I',"Insert");
                    chelp(0,'D',"Delete");
                    buf=attachmenu(length);
                    if(buf==CTRL_I)

```

```

        msg_reverse(21,"[write [path]V[file name] and press ENTER]");
        length= insert(length);
        if(buf!=CTRL_X)
            buf='B';
    }
    if(buf==CTRL_D)
    {
        length=delete(length);
        if(buf!=CTRL_X)
            buf='B';
    }
    chelp(1,'I',"");
    chelp(1,'D',""); break;
case 4 :   hv("hcancel");break;
}
endby=0;
if(buf==CTRL_S||wantexit==2)
{
    now=confirm("Do you want to send ?(y/n)",now,oldnow);
    wantexit=0;
    endby=1;
}
if(buf==CTRL_X||wantexit==1)
{
    now=confirm("Do you want to exit ? (y/n)",now,oldnow);
    wantexit=0;
    endby=2;
}
else if(now!=4)
{
    oldnow=now;
    now=ananext(buf,now);
}
else

```

```

    {
        now=oldnow;
    }

    refresh();
}while(now!=0);

endwin();
if(endby==2)
    printf("\n\n EXIT homework sending file \n\n");
if(endby==1)
    printf("\n\n sending homework file \n\n");
system("stend");
}

```



การเลื่อนห้องเรียน

โปรแกรมย่อยที่เรียกใน main() ทั้งหมดนี้ได้บรรยายไว้ในหัวข้อการนัดชั้นเรียนแล้วแตกต่างกันเพียงในส่วนการบริหารขั้นตอนดังแสดงข้างล่างนี้

ลักษณะพิเศษของ main คือจะทำงานตามคำสั่งที่เข้ามาในตอนเรียกโปรแกรมคือสามารถเลือกได้ว่าต้องการใช้อาจารย์เป็นหลักหรือใช้นักเรียนเป็นหลักในการทำงานเกี่ยวกับฐานข้อมูล

```
main(argc,argv)
int argc;
int *argv[2];
{ FILE *ft,*fs;
  char *cbuf;
  char *delc;
  char *sb;
  char *week;
  int stdorteac;
  int tnum;
  int maxt,maxs;
  struct table t[MAX_TEACH], *sbuf;
  struct table s[MAX_STD];
  int i,o,v;
  char *member[20];
  char tcancel[MAX_MENU1];
  system("smod");
  initscr();

  stdorteac=argc;
  oldcheckexitmenu1=CTRL_E;
  checkterm(); /* check term which program can run normally */
  move(12,0);
  chelp(3,'O',"writeOut");
  chelp(0,'H',"use Help");

  msg_reverse(MSG_LINE,"[Opening file teac and std]");
  ft=fopen("teac","r");
```

```

fs=fopen("std","r");
cbuf=readf(ft);
maxt=atoi(cbuf);

for(o=0;o<maxt;o++)
{
    sbuf=readteach(ft);
    for(i=0;i<35;i++)
        t[o].name[i]=sbuf->name[i];
    for(v=0;v<(sbuf->number);v++)
    {
        t[o].d[v]=malloc(MAX_READ_ONCET);
        for(j=0;j<100;j++)
        {
            t[o].d[v][j]=sbuf->d[v][j];
        }
    }
    t[o].number=sbuf->number;
}
cbuf=readf(fs);
maxs=atoi(cbuf);

for(o=0;o<maxs;o++)
{
    sbuf=readteach(fs);
    for(i=0;i<35;i++)
        s[o].name[i]=sbuf->name[i];
    for(v=0;v<(sbuf->number);v++)
    {
        s[o].d[v]=malloc(100);
        for(i=0;i<100;i++)
        {
            s[o].d[v][i]= sbuf->d[v][i];
        }
    }
}

```

```

    }

    s[o].number=sbuf->number;

}

fclose(fs);
fclose(ft);

msg_reverse(MSG_LINE,"[already load file std and teac]");
/* MALLOC PART*/
for(i=0;i<20;i++)
member[i]= malloc(50);
delc=malloc(MAX_MENU1);
sb=malloc(40);
week=malloc(40);

/* ++++++if select TEACHER DOMAIN+++++*/

checkexitmenu1=CTRL_E;
if(stdorteac==1)
{
    do{
        if(checkexitmenu1==CTRL_E)
        {
            wrongkey=0;
            clearpage(3,20);
            showt(t,maxt,s,maxs,sb);
            current_item1=0;
            current_itemt=0;
            if(checkexitmenu1!=CTRL_X)
                checkexitmenu1=CTRL_F;
            current_window_M=1;
        }

        if(checkexitmenu1!=CTRL_X&&checkexitmenu1!=CTRL_E)
            if(checkexitmenu1==CTRL_F)
            {

```

```

        wrongkey=0;
        current_item1=0;
        tnum=findtec(sb,t,maxt,tcancel);
        current_item1=0;
        current_window_M=2;
        if(checkexitmenu1==CTRL_M)
            checkexitmenu1=CTRL_L;
    }

    if(checkexitmenu1!=CTRL_X&&checkexitmenu1!=CTRL_E)
        if(checkexitmenu1==CTRL_L||checkexitmenu1==CTRL_D)
        {
            wrongkey=0;
            current_item1=0;
            choosemember(tnum,tcancel,s,maxs,t,maxt,member,delc);
            current_window_M=3;
            if(checkexitmenu1==CTRL_M)
                checkexitmenu1=CTRL_Z;
        }

    if(checkexitmenu1!=CTRL_X&&checkexitmenu1!=CTRL_E&&checkexitmenu1!=CTRL_F&&checkexitmenu1!=CTRL_D)

        if(checkexitmenu1==CTRL_Z)
        {
            wrongkey=0;
            current_item1=0;
            chelp(0,'L','deLete');
            strcpy(week,whatweek());
            if(checkexitmenu1==CTRL_M)
                confirm("Do you Want to send Cancel-class (y/n) :");
        }

        if(checkexitmenu1==CTRL_H)
        {
            hv("hcancel");

```

```

    }

    wrongkey++;

    if(wrongkey>1)

        checkexitmenu1=oldcheckexitmenu1;

    if(checkexitmenu1==CTRL_X)

        confirm("Do you Want to Exit Cancel-class now? (y/n) :");

    } while(checkexitmenu1!=CTRL_X&&checkexitmenu1!=CTRL_M);

}

/*+++++++if select STUDENT DOMAIN+++++++*/

else if(stdorteac==2)
{
    do{
        if(checkexitmenu1==CTRL_E)
        {
            wrongkey=0;
            clearpage(3,20);
            showt(s,maxs,t,maxt,sb);
            current_item1=0;
            current_itemt=0;

            if(checkexitmenu1!=CTRL_X)

                checkexitmenu1=CTRL_F;

            current_window_M=1;

        }

        if(checkexitmenu1!=CTRL_X&&checkexitmenu1!=CTRL_E)

            if(checkexitmenu1==CTRL_F)

            {

                wrongkey=0;

                current_item1=0;

                tnum=findtec(sb,s,maxs,tcancel);

                current_item1=0;

                current_window_M=2;

```

```

        if(checkexitmenu1==CTRL_M)
            checkexitmenu1=CTRL_L;
    }

    if(checkexitmenu1!=CTRL_X&&checkexitmenu1!=CTRL_E)
        if(checkexitmenu1==CTRL_L||checkexitmenu1==CTRL_D)
        {
            wrongkey=0;
            current_item1=0;
            choosemember(tnum,tcancel,t,maxt,s,maxs,member,delc);
            current_window_M=3;
            if(checkexitmenu1==CTRL_M)
                checkexitmenu1=CTRL_Z;
        }

    if(checkexitmenu1!=CTRL_X&&checkexitmenu1!=CTRL_E&&checkexitmenu1!=CTRL_F&&checkexitmenu1!=CTRL_D)
        if(checkexitmenu1==CTRL_Z)
        {
            wrongkey=0;
            current_item1=0;
            chelp(0,'L','deLeTe');
            strcpy(week,whatweek());
            if(checkexitmenu1==CTRL_M)
                confirm("Do you Want to send Cancel-class (y/n) :");
        }

    if(checkexitmenu1==CTRL_H)
    {
        hv("hcancel");
    }

    wrongkey++;
    if(wrongkey>1)
        checkexitmenu1=oldcheckexitmenu1;

    if(checkexitmenu1==CTRL_X)

```

```

        confirm("Do you Want to Exit Cancel-class now? (y/n) :");
    } while(checkexitmenu1!=CTRL_X&&checkexitmenu1!=CTRL_M);
}
endwin();
if(checkexitmenu1==CTRL_M)
{
    if(stdorteac==1)
    {
        makedraft(week,tcancel,t[tnum].name,member);
    }else
        makedraft(week,tcancel,s[tnum].name,member);

    printf("\n\n Cancel Class finish. \n\n");
}
else printf("\n\n Cancel Class Exit. \n\n");
}
}

```

นัดเวลาเรียน (make up class)

ประกอบด้วยโปรแกรมย่อยที่ทำเมนูแต่ละขั้นตอนโดยทำตามลำดับและใช้เทคนิคการทำแบบเดินหน้า ถ้าจะถอยหลังจะลบข้อมูลที่ทำหลังจากที่ถอยทั้งหมด(undo)การออกแบบโปรแกรมจึงเห็นการเปลี่ยนขั้นตอนการทำโดยใช้ปุ่มควบคุมแทนที่จะใช้การเลื่อนลูกขรขึ้นลงธรรมดาเพราะการใช้อักษรเลื่อนขึ้นลงมีผลในการสร้างความผิดพลาดในการย้อนกลับโดยไม่ตั้งใจเพราะสามารถย้อนหลังได้โดยง่ายแต่การใช้ปุ่มควบคุมนั้นจะสร้างความระมัดระวังที่สูงกว่า

โปรแกรมหลัก

curses

```
#include <curses.h>
```

กำหนดขนาดฐานข้อมูลสูงสุดที่ต้องการใช้

```
#define MAX_TEACH 40
#define MAX_STD 40
#define MAX_LEA 14
#define MAX_READ_ONCET 100
#define NAME 35
```

โครงสร้างของฐานข้อมูล

```
struct table{
    char name[NAME];
    char *d[MAX_LEA];
    int number;
};
```

อ่านเพิ่มข้อมูลฐานข้อมูลให้อยู่ในหน่วยความจำเพื่อใช้งาน

```
/* readteach: function
   return  :struct table
   affect   :move pointer to end of struct
   do       :read struct of teach and std for table form file teac and std only
*/
struct table *readteach(fp)
FILE *fp;
```

```

{ struct table *t;
  int i,counter=1;
  char *buf;
  buf=readf(fp);

  for(i=0;i<NAME;i++)
  {
    if(buf[i]!='\n'&&buf[i]!='\r')
      t->name[i]=buf[i];
    else t->name[i]=0;
  }
  while(buf[0]!='!')
  {
    buf=readf(fp);
    if(buf[0]!='!')
    {
      i=1;
      t->d[counter]=malloc(MAX_READ_ONCET);
      do{
        i++;
        if(buf[i]!='\n'&&buf[i]!='\r')
          t->d[counter][i]=buf[i];
        else
          t->d[counter][i]=0;
      }while(buf[i]!=0);
    }
    counter++;
  }
  t->number=counter-1;
  return t;
}

```

ส่วนบริหารการเปลี่ยนชั้นของเมนู (main)

```

/* global : use because stack overflow in memory */
char *sbweek;
int wrongkey=0;
main(argc,argv) /* MAIN */
int argc;
int *argv[2];
{
    FILE *ft,*fs;
    char *cbuf;
    char *delc;
    char *sb;
    int stdorteac;
    int trum;
    int maxt,maxs;
    struct table t[MAX_TEACH], *sbuf;
    struct table s[MAX_STD];
    int i,o,v;
    char *member[20];
    char tcancel[MAX_MENU1];
    char *makex;

    savetty();
    system("mesg -n");
    system("smod");
    initscr();

    stdorteac=argc;

    checkterm(); /* check term which program can run normally */
    move(12,0);
    chelp(3,'O',"writeOut");
    chelp(3,'X',"eXit");

```

```

chelp(0,'H','use Help');

msg_reverse(MSG_LINE,['Opening file teac and std']);

ft=fopen("teac","r");

fs=fopen("std","r");

cbuf=readf(ft);

maxt=atoi(cbuf);

for(o=0;o<maxt;o++)
{
sbuf=readteach(ft);
for(i=0;i<35;i++)
    t[o].name[i]=sbuf->name[i];
    for(v=0,v<(sbuf->number);v++)
    {
        t[o].d[v]=malloc(MAX_READ_ONCET);
        for(i=0;i<100;i++)
        {
            t[o].d[v][i]=sbuf->d[v][i];
        }
        t[o].number=sbuf->number;
    }
}

cbuf=readf(fs);

maxs=atoi(cbuf);

for(o=0;o<maxs;o++)
{
sbuf=readteach(fs);
for(i=0;i<35;i++)
    s[o].name[i]=sbuf->name[i];
    for(v=0,v<(sbuf->number);v++)
    {
        s[o].d[v]=malloc(100);
        for(i=0;i<100;i++)

```

```

        {
            s[o].d[v][i]= sbuf->d[v][i];
        }
    }

    s[o].number=sbuf->number;

}

fclose(fs);

fclose(ft);

msg_reverse(MSG_LINE,"already load file std and teac");

/* MALLOC PART*/

for(i=0;i<20;i++)
member[i]= malloc(50);
delo=malloc(MAX_MENU1);
sb=malloc(40);
makex=malloc(60);
sbweek=malloc(40);

/* ++++++if select TEACHER DOMAIN+++++*/

checkexitmenu1=CTRL_E;
if(stdorteac==1)
{
do{
    chelp(1,"W","");
    chelp(1,"K","");
    if(checkexitmenu1==CTRL_E)
    {
        wrongkey=0;
        clearpage(4,20);
        showt(t,maxt,s,maxs,sb);
        current_item1=0;
        current_itemt=0;
        if(checkexitmenu1!=CTRL_X)
            checkexitmenu1=CTRL_F;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        current_window_M=1;
    }

    if(checkexitmenu1!=CTRL_X&&checkexitmenu1!=CTRL_E)
        if(checkexitmenu1==CTRL_F)
        {
            wrongkey=0;
            clearpage(4,20);
            tnum=findtec(sb,t,maxt,tcancel);
            current_item1=0;
            current_window_M=2;

            if(checkexitmenu1==CTRL_M)
                checkexitmenu1=CTRL_L;
        }
    if(checkexitmenu1!=CTRL_X&&checkexitmenu1!=CTRL_E)
        if(checkexitmenu1==CTRL_L||checkexitmenu1==CTRL_D)
        {
            wrongkey=0;
            current_item1=0;
            choosemember(tnum,tcancel,s,maxs,t,maxt,member,delc);
            current_window_M=3;
            if(checkexitmenu1==CTRL_M)
                checkexitmenu1=CTRL_W;
        }

    if(checkexitmenu1!=CTRL_X)
        if(checkexitmenu1==CTRL_W)
        {
            wrongkey=0;
            chelp(1,'A','');
            chelp(1,'D','');
            chelp(0,'L','delete');
            strcpy(sbweek,whatweek());
            if(checkexitmenu1==CTRL_M)
                checkexitmenu1=CTRL_K;
        }

```

```

    }

if(checkexitmenu1!=CTRL_X&&checkexitmenu1!=CTRL_E&&checkexitmenu1!=CTRL_F&&checkexitmenu1!=CTRL_D)

    if(checkexitmenu1==CTRL_K)
    {
        wrongkey=0;
        clearpage(9,20);
        strcpy( makeox,makedclass(member,t,maxt,s,maxs));
        if(checkexitmenu1==CTRL_M)
            confirm("Do you sure accept to makeup-class (y/n)?");
        if(checkexitmenu1!=CTRL_M&&checkexitmenu1!=CTRL_X)
            clearpage(9,20);
    }
    if(checkexitmenu1==CTRL_H)
        hv("hmakeup");

    if(checkexitmenu1==CTRL_X)
        confirm("Do you want to exit makeup-class (y/n)?");
        wrongkey++;

    if(wrongkey>1)
        checkexitmenu1=oldcheckexitmenu1;

    if(wrongkey>2)
        checkexitmenu1=CTRL_K;

    } while(checkexitmenu1!=CTRL_X&&checkexitmenu1!=CTRL_M);
}

/*+++++++if select STUDENT DOMAIN+++++++*/

```

```

{
do{
    chelp(1,'W',' ');
    chelp(1,'K',' ');
    if(checkexitmenu1==CTRL_E)
    {
        {
            wrongkey=0;
            clearpage(3,20);
            showt(s,maxs,t,maxt,sb);
            current_item1=0;
            current_itemt=0;
            if(checkexitmenu1!=CTRL_X)
                checkexitmenu1=CTRL_F;
            current_window_M=1;
        }
        if(checkexitmenu1!=CTRL_X&&checkexitmenu1!=CTRL_E)
            if(checkexitmenu1==CTRL_F)
            {
                wrongkey=0;
                clearpage(4,20);
                tnum=findtec(sb,s,maxs,tcancel);
                current_item1=0;
                current_window_M=2;
                if(checkexitmenu1==CTRL_M)
                    checkexitmenu1=CTRL_L;
            }
        if(checkexitmenu1!=CTRL_X&&checkexitmenu1!=CTRL_E)
            if(checkexitmenu1==CTRL_L||checkexitmenu1==CTRL_D)
            {
                wrongkey=0;
                current_item1=0;
                choosemember(tnum,tcancel,t,maxt,s,maxs,member,delc);
                current_window_M=3;
                if(checkexitmenu1==CTRL_M)

```

```

        checkexitmenu1=CTRL_W;
    }
    if(checkexitmenu1!=CTRL_X)
        if(checkexitmenu1==CTRL_W)
        {
            wrongkey=0;
            chelp(1,'A',' ');
            strcpy(sbweek,whatweek());
            if(checkexitmenu1==CTRL_M)
                checkexitmenu1=CTRL_K;
        }

if(checkexitmenu1!=CTRL_X&&checkexitmenu1!=CTRL_E&&checkexitmenu1!=CTRL_F&&checkexitmenu1!=CTRL_D)
    if(checkexitmenu1==CTRL_K)
    {
        wrongkey=0;
        clearpage(9,20);
        strcpy( makex,makeclass(member,s,maxs,t,maxt));
        if(checkexitmenu1==CTRL_M)
            confirm("Do you sure accept to makeup-class (y/n)");
        if(checkexitmenu1!=CTRL_M&&checkexitmenu1!=CTRL_X)
            clearpage(9,20);
    }

    if(checkexitmenu1==CTRL_H)
        hv("hmakeup");

    wrongkey++;
    if(wrongkey>1)
        checkexitmenu1=oldcheckexitmenu1;
    if(checkexitmenu1==CTRL_X)
        confirm("Do you want to exit makeup-class (y/n)");

```

```

while(checkexitmenu1!=CTRL_X&&checkexitmenu1!=CTRL_M);

}

endwin();

if(checkexitmenu1==CTRL_M)
{
    if(stdorteac==1)
        makedraft(sbweek,makex,tcancel,tjtnum).name,member);
    else
        makedraft(sbweek,makex,tcancel,sjtnum).name,member);
    printf("\n\n Make-up Class finish \n\n",makex,tcancel,tjtnum,member[0]);
}
else
    printf("\n\n Make-up Class exit \n\n");
}

```

ส่วนการให้ชื่อที่ใช้เป็นคีย์(key)

```

int showt(t,maxt,s,maxs,sb)
struct table t[MAX_TEACH];
int maxt;
struct table s[MAX_STD];
int maxs;
char sb[40];
{
    int matchok=1;
    int o,buf,i,cin;
    int x,y,xbuf;
    int selectteac=1,selectstd=1;
    char *tablet[MAX_TEACH],*tables[MAX_STD];
    /*_____*/
}

```

```

line_reverse(0,"Cancel class by member");

o=-1;

/* tranfer from struct to array,for sbnw in membe!*/
for(i=1;i<=maxt;i++)
{
    tablet[i]=malloc(NAME);
    for(o=0;o<NAME;o++)
        tablet[i][o]=t[i-1].name[o];
}
do{
    msg_reverse(21,"1.put your group name or your name ENTER 2).CTRL_A");
    /*-----read key PART-----*/

attrset(A_REVERSE);
attron(A_REVERSE);
move(2,2);
addstr("Member name :");
refresh();
echo();
refresh();

    chelp(1,'L','');
    chelp(1,'D','');
    chelp(1,'F','');
    chelp(1,'E','');
    chelp(3,'H',"Help");
    chelp(3,'X',"eXit");
    chelp(0,'A',"Add Name");
    move(2,18);
    readline(sb,40);

/*-if user want to select from menu-----*/

if(sb[0]==CTRL_A)

```

```

{ chelp(1,'A','');
  chelp(3,'E','Edit name');
  chelp(0,'M','Accept');
  move(4,17));
  strcpy(sb,selects(tablet,maxt+1,6));
/*—if he refuse to use it then exit abnormally————*/
if(checkexitmenu1!=13)
  sb[0]=0;

}

/*-refresh screen————*/
if(sb[0]==CTRL_X)
  checkexitmenu1=CTRL_X;
matchok=testmatching(sb,t,maxt);
if(sb[0]==CTRL_H)
  hv("hcancel");
else if(checkexitmenu1==CTRL_H)
  hv("hcancel");
else
{
  move(2,18);
  addstr("
");
  move(2,18);
  addstr(sb);
  refresh();
}

}while(matchok!=0&&checkexitmenu1!=CTRL_X);
/* while if not cmp yes and not want to exit*/
refresh();

/*————*/

attroff(A_REVERSE);
move(2,2);

```

```

addstr("Member name :");
refresh();
/*analyze vacation : */
}

testmatchind(sb,t,maxt)
char sb[40];
struct table t[MAX_TEACH];
int maxt;
{ int o,buf;
  for(o=0;o<maxt;o++)
  { buf=strcmp(sb,t[o].name);
    if(buf==0)
    { return 0;
    }
  }
  return 1;
}

```

เลือกวิชาที่ต้องการสมัคร

```

findtec(sb,t,maxt,std)
char sb[40];
struct table t[MAX_TEACH];
int maxt;
char std[MAX_MENU1];
{ int o,buf,selecttec;
  current_item1=0;
  clearpage(3,20);
  chelp(1,'L',' ');
  chelp(1,'D',' ');
  chelp(1,'F',' ');
  chelp(1,'A',' ');
  chelp(3,'E','Edit name');

```

```

chelp(0,'M','Accept');
move(4,2);
attron(A_REVERSE);
addstr("Subject :");
refresh();
move(4,12);
for(o=0;o<maxt;o++)
{
    buf=strcmp(sb,t[o].name); /*compare old name*/
    if(buf==0) /* if yes*/
    {
        strcpy(std, ts[t[o].d,t[o].number,10));
        selectteac=o;
        o=maxt;
    }
}
refresh();
move(4,2);
attroff(A_REVERSE);
addstr("Subject :");
refresh();
if(checkexitmenu1!=CTRL_F)
{
    clearpage(3,20);
    move(4,2);
    addstr("Subject :");
    add_str(std);
}
return selectteac;
}
/*_____*/
#include <ctype.h>

```

```

#include <string.h>
#include <strings.h>

#define MAX_MENU1 200 /* max of array on menu */

int current_itemt; /* globol reserve for this menu1.h */

/*redeclare
int extrakey;
int longest_1;
*/

/* select : function
return : string that selected
do : receive user in put character or cross to select menu
* : extra menu use like menu1
*/

/* redeclare checkexitmenu1 from menu1.h */
char *ts(menu, menu_length,length)
char *menu[MAX_MENU1];
int menu_length;
int length;
{
    int hasesubject[7];
    int c=0;
    int x,y,i;
    int xb,yb;

    longest_1=0;
    longest_1=40;

    savetty(); /* save tty of system*/
    cbreak();nonl();noecho();standend();

    /* set use Ctrl-break and no echo to getch */
    keypad(stdscr,TRUE);

    /* enable keypad to use extra keyboard*/
    msg_reverse(21,"Press UP DOWN LEFT RIGHT key");
    makehasesubject(menu,menu_length,hasesubject);
    getyx(stdscr,y,x); /* from where */
    for(i=0;i<length+2;i++)

```

```

    {
        move(y+1,x);
        addstr(" ");
    }

do{
    move(y,x);
    /* goto where */
    current_itemt=checkkeyf(menu_length,menu,current_itemt,hassubject);
    /* check current key to translate mean*/
    pumenu(menu,menu_length,length,current_itemt);
    /* display menu*/
    refresh();
    c=readch();
    while(c!='\n'&&c!='V'&&!(c>=1&&c<=26)); /* exit value*/
    resetty(); /* return tty environment */
    oldcheckexitmenu1=checkexitmenu1;
    checkexitmenu1=c;
    return menu[current_itemt+1]; /* return string selected*/
}

makehassubject(menu,menu_length,hassubject)
char *menu[MAX_MENU1];
int menu_length;
int hassubject[7]; /* max subject is 6 */
{
    char day[7][3];
    int i,o;
    char buff[10];

    strcpy(day[0],"sun");
    strcpy(day[1],"mon");
    strcpy(day[2],"tue");
    strcpy(day[3],"wed");
    strcpy(day[4],"thu");

```

```

strcpy(day[5],"fri");
strcpy(day[6],"sat");

for(i=0;i<7;i++)
    hasesubject[i]=0;

for(i=1;i<menu_length;i++)
{
    strcpy(buf,search(menu[i],0));
    for(o=0;o<7;o++)
    {
        if(strcompare(buf,day[o],3)==0)
            hasesubject[o]++;
    }
}
}

/* checkkey : function
return : mean of key for menu selecting
do : analize key (c)
*/
int checkkey(c,menu_length,menu,current_itemt,hasesubject)
int c;
int menu_length;
int current_itemt;
char *menu[MAX_MENU1];
int hasesubject[7];
{
    if((c==KEY_UP||c=='A'&&extrakey==1))
    {
        current_itemt=cutc(hasesubject,1);
        msg_reverse(21,"Press UP DOWN LEFT RIGHT key");
    }

    else if(current_itemt<menu_length-firstsubject(hasesubject,2)-
1&&(c==KEY_DOWN||c=='B'&&extrakey==1))
    {
        current_itemt=cutc(hasesubject,2);
    }
}

```

```

        msg_reverse(21,"[Press UP DOWN LEFT RIGHT key]");
    }

    else if((current_itemt<menu_length-2)&&(c==KEY_RIGHTlc=='C'&&extrakey==1))
    {

        current_itemt++;

        msg_reverse(21,"[Press UP DOWN LEFT RIGHT key]");

    }

    else if((current_itemt>0)&&(c==KEY_LEFTlc=='D'&&extrakey==1))
    {
        current_itemt--;

        msg_reverse(21,"[Press UP DOWN LEFT RIGHT key]");

    }

    else
    {

        current_itemt=current_itemt;

        msg_reverse(21,"[Press UP DOWN LEFT RIGHT for selecting]");

    }

    return current_itemt;

    /* return number of line is selected*/

}

firstsubject(s,i)
int s[7];
int i;
{
    int o;

    o=0;

    if(i==1)
    {

        while(s[o]==0)

            o++;

        return s[o];

    }

    o=6;

    if(i==2)
    {

```

```

        while(s[o] == 0)
            o--;
        return s[o];
    }
}

auto(havesubject, updown)
int havesubject[7];
int updown;
{
    int i=0, o, counter=0;
    while(i <= current_itemt)
    {
        o = havesubject[counter];
        i = i + o;
        counter++;
    }
    if(updown == 2)
        return (i);
    if(updown == 1)
        return current_itemt - havesubject[counter-2];
}

/* pumenut : procedure
   return : none
   do : print menu1 for selects
*/

pumenut(menu, menu_length, length, current_itemt)
char *menu[MAX_MENU1];
int menu_length;
int length;
int current_itemt;
{
    int i=0, o, v, e, i1;
    int counter=0;
    int linebegin;

```

```

int x,y;

char day[7][3];

char *checkday;

char bufm[MAX_MENU1];

strcpy(day[0],"sun");

strcpy(day[1],"mon");

strcpy(day[2],"tue");

strcpy(day[3],"wed");

strcpy(day[4],"thu");

strcpy(day[5],"fri");

strcpy(day[6],"sat");

checkday=malloc(10);

getyx(stdscr,y,x);

makeborder(y,x);

for(i=1;i<menu_length;i++)
{
    strcpy(checkday,search(menu[i],0));

    for(o=0;o<7;o++)
    {
        if(strcompare(day[o],checkday,3)==0)
        {
            strcpy(bufm,search(menu[i],1));

            v=genx(bufm);

            e=genend(bufm);

            if(i==(current_itemt+1))
            {

                attron(A_REVERSE);

                move(y+10,x);

                add_str(menu[i]);

            }

            else

                attroff(A_REVERSE);

            move(y+o+1,x+5+v);

```

```

        strcpy(bufm,search(menu[i],2));

        puts(subject(v,e,bufm);

    }

}

refresh();

}

add_str(c)
char c[MAX_MENU1];
{
    int counter=0;
    int y,x;
    getyx(stdscr,y,x);
    addstr(" ");
    move(y,x);
    while(c[counter]!=0)
    {
        if(c[counter]!='_')
            addch(c[counter]);
        else addch(" ");
        counter++;
    }
}

puts(subject(v,e,bufm)
int v;
int e;
char bufm[10];
{
    int check0=1,j;

    for(j=v;j<e;j++)
    {
        if(bufm[j-v]==0)
            check0=0;
    }
}

```

```

        if(check0==1)
        {
            addch(bufm[i-v]);
        }
        else addch('.');
    }
}

```

```

genend(bufm)

```

```

char bufm[10];

```

```

{ char buf[10];

```

```

    char b;

```

```

    int checkf;

```

```

    int i,o;

```

```

    int counter;

```

```

    int r1,r2;

```

```

    int stobuf;

```

```

    i=0;

```

```

    refresh0;

```

```

    while(bufm[i]!='-')

```

```

    {

```

```

        i++;

```

```

    }

```

```

    i++;

```

```

    counter=0;

```

```

    checkf=0;

```

```

    while(bufm[i]!='.' && checkf<3)

```

```

    {

```

```

        b=bufm[i];

```

```

        buf[counter]=b;

```

```

        i++;

```

```

        checkf++;

```

```

        counter++;

```

```

    }

```

```

    buf[counter]=0; /* for hour part */

```

```

atoibuf=atoi(buf);
r1=(atoibuf-7)*4; /*for 7:00*/
/* detect fault path*/
if(checkf==3)
    return r1;
i++;
counter=0;
refresh();
while(bufm[i]!=0)
{
    b=bufm[i];
    buf[counter]=b;
    i++;
    counter++;
}
refresh();
buf[counter]=0;
atoibuf=atoi(buf);
atoibuf++;
r2=atoibuf/15; /* for minite part*/
r1=r1+r2;
refresh();
return r1 ;/* for 15 minite*/
)

```

```

genx(bufm)

```

```

char bufm[10];

```

```

{
    char buf[3];
    int bufatoi;
    int checkf;
    int i,counter;
    int r1,r2;
    i=0;
    counter=0;
    check=0;

```

```

while(bufm[i]!='.' && checkf<3)
{
    buf[counter]=bufm[i];
    i++;
    counter++;
    checkf++;
}

buf[counter]=0; /* for hour part*/
r1=(atoi(buf)-7)*4; /*for 7:00*/
/*detect error path*/
if(checkf==3)
    return r1;
i++;
counter=0;
checkf=0;
while(bufm[i]!='.' && checkf<3)
{
    buf[counter]=bufm[i];
    i++;
    counter++;
    checkf++;
}

buf[counter]=0;
bufatoi=atoi(buf); /* for minite part*/
bufatoi++;
r2=bufatoi/15;
r1=r1+r2;
return r1 ; /* for 15 minite*/
}

```

```
strcompare(c,s,i)
```

```
char *c;
```

```
char *s;
```

```
int i;
```

```
{ int o;
```

```

int status=0;

for(o=0;o<i;o++)

    if(d[o]!=s[o])

        status=1;

return status;

}

makeborder(y,x)

int y;
int x;
{
    int i;

    move(y,x);
    getyx(stdscr,y,x);
    attron(A_REVERSE);
    addstr("  7:00  9:00  11:00  13:00  15:00  17:00  19:00  21:00");
    move(y+1,x);
    addstr(" Sun ");
    move(y+2,x);
    addstr(" Mon ");
    move(y+3,x);
    addstr(" Tue ");
    move(y+4,x);
    addstr(" Wed ");
    move(y+5,x);
    addstr(" Thu ");
    move(y+6,x);
    addstr(" Fri ");
    move(y+7,x);
    addstr(" Sat ");
    move(y+8,x);
    addcht(' ',66);
    for(i=0;i<9;i++)
    {
        move(i+y,66+x);

```

```

        addch(" ");
    }

    refresh();
}

addch(c,i)
char c;
int i;
{ int o;
  for(o=0;o<i;o++)
    addch(c);
  refresh();
}

```

ส่วนทำการเลือกสัปดาห์ที่การนัดเรียนมีผล

```

/* whatweek : function
   return   : week that selected
   do       : suggest user what week they want to cancel or make up class
*/
char *whatweek()
{
    char *week[60];
    char *weeks;
    int x,y;
    int i;
    char *c;

    chelp(1,'D',' ');
    move(8,2);
    attron(A_REVERSE);
    addstr("What week  :");
    refresh();

```

```

    for(i=0;i<60;i++)

```

```

week[i]=malloc(40);
weeks=malloc(40);
c=malloc(3);
strcpy(week[1],"This week");
strcpy(week[2],"Next week");
strcpy(weeks,"Next____week");
for(i=3;i<60;i++)
{
    strcpy(c,itos(i-1));
    weeks[5]=c[0];

    if(c[1]!='0')
        weeks[6]=c[1];
    strcpy(week[i],weeks);
}
move(8,17);
strcpy(weeks,selects(week,60,6));
clearpage(7,20);
move(8,2);
attroff(A_REVERSE);
addstr("What week :");
addstr(" ");
add_str(weeks);
refresh();

return weeks;
}

```

ส่วนเมนูเลือกสมาชิกที่ต้องการส่งจดหมาย(mail)ให้

```

/*
choosemember : function

```

```

act : third menu

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return      : member that deleted
*/
int current_window_M=0;
char *choosemember(tnum,tcancel,t,maxt,s,maxs,member,dclc)
int tnum;
char tcancel[MAX_MENU1];
struct table t[MAX_TEACH];
int maxt;
struct table s[MAX_STD];
int maxs;
char *member[20];
{ int i,m,counter;
  int ok=0;
  static char memberbuf[20];
  clearpage(5,20);
  if(current_window_M!=3)
    for(i=0;i<20;i++)
      memberbuf[i]=' ';
  finds(tnum,tcancel,t,maxt,s,maxs,member);
  chelp(3,'F',"Find Sub");
  chelp(0,'D',"rmv member");
  chelp(1,'L',"");

  for(i=1;i<member[0][0]+1;i++)
    if(memberbuf[i]=='*')
      member[i][0]='*';
      move(6,2);
      attron(A_REVERSE);
      addstr("Staff && Std :");
      refresh();
      move(6,17);
      strcpy(dclc,selects(member,member[0][0]+1,7));

```

```

if(checkexitmenu1==CTRL_D)
{
    current_item1++;
    if(memberbuf[current_item1]!='*')
        memberbuf[current_item1]='*';
    else
        memberbuf[current_item1]=' ';
}

clearpage(5,20);
move(6,2);
attroff(A_REVERSE);
addstr("Staff && Std :");
addstr(" ");

for(j=1;j<=member[0][0];j++)
{
    if(member[j][0]!='*')
    {
        if(ok==1&&j>1)
            addstr(",");
        addstr(member[j]);
        ok=1;
    }
}

refresh();
return memberbuf;
}

char *search(tcancel,point)
char tcancel[MAX_MENU1];
int point;
{ int i=0,counter=0,icount=0;
  char st[100];
  if(point<0)
  { addstr("Error [2] :May occur core dump protection ");
    exit(0);
  }

```

```

)
while(counter<point&&tcancel[i]!=0)
{ if(tcancel[i]!='_')
{ while(tcancel[i]!='_')
{ i++; }
}
if(tcancel[i]==0&&tcancel[i]!='_')
{ while(tcancel[i]!='_')
{ i++;}
}
counter++;
}
while(tcancel[i]!='_')
{ st[icount]= tcancel[i];
i++;
icount++;
}
st[icount]=0;
return st;
}

findsnumber(s,maxs,st)
struct table s[MAX_STD];
int maxs;
char st[100];
{ int counter=0;

while(strcmp(s[counter].name,st)!=0&&(counter<=maxs))
counter++;
if(counter>maxs)
return -1;
else
return counter;
}

```

```

Error(c)
char c[100];
{
    msg_reverse(21,c);
    refresh();
    endwin();
    exit(0);
}

char *findt(s,snumber,tcancel)
struct table s[MAX_STD];
int snumber;
char tcancel[MAX_MENU1];
{
    int i=1;
    int check=0;
    char buf1[100],buf2[100],buf3[100],buf4[100],buf5[100],buf6[100];
    if(snumber<0)
    {
        Error("Error [2] : protection core dump ");
    }
    while(i<=s[snumber].number)
    {
        check=0;
        refresh();
        strcpy(buf1,search(tcancel,0));
        strcpy(buf2,search(s[snumber].d[i],0));
        if(strcmp(buf1,buf2)==0)
        {
            check=1;
        }
        if(check==1)
        {
            strcpy(buf3,search(tcancel,1));
            strcpy(buf4,search(s[snumber].d[i],1));
            if(strcmp(buf3,buf4)==0)
            {
                check=2;
            }
        }
    }
}

```

```

        if(check==2)
        {
            strcpy(buf5,search(tcancel,2));
            strcpy(buf6,search(s[snumber].d[i],2));
            if(strcmp(buf5,buf6)==0)
                check=3;
        }
    }
    if(check==3)
        return (s[snumber].d[i]);
    i++;
}
return("Error [1]: No anyone match");
}

/* receive tcancel from open.c and search student name to recheck tname */
finds(tnum,tcancel,s,maxs,t,maxt,member)
int tnum;
char tcancel[MAX_MENU1];
struct table s[MAX_STD];
int maxs;
struct table t[MAX_TEACH];
int maxt;
char *member[20];
{
    char st[100];
    char checkt[MAX_MENU1];
    char outst[10][100];
    char cutt[10][100];
    int counter=0;
    int outstcounter=0;
    int cuttcounter=0;
    int i=0;

```

```

int snumber;

for(i=0;i<20;i++)

member[i]=malloc(50);

strcpy(st,search(tcancel,4)); /* now I use 4 mean stdname*/

i=0;
cutstcounter=0;
do{
    counter=0;
    while(st[i]!=44&&st[i]!=0)
    {
        cutst[cutstcounter][counter]=st[i];
        counter++;
        i++;
    }
    cutst[cutstcounter][counter]=0;
refresh();
    snumber=findsnumber(s,maxs,cutst[cutstcounter]);
    strcpy(checkt,findt(s,snumber,tcancel));
    i++;
    cutstcounter++;
} while(st[i-1]!=0);

strcpy(st,search(checkt,4));
i=0;
cuttcounter=0;
do{
    counter=0;
    while(st[i]!=0&&st[i]!=44)
    { cutt[cuttcounter][counter]=st[i];
        counter++;
        i++;
    }

    cutt[cuttcounter][counter]=0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        cuttcounter++;

        i++;
    }while(st[j-1]!=0);

    member[0][0]=cutstcounter+cuttcounter;
    i=1;
    counter=0;

    while(counter<cutstcounter)
    {
        strcpy(member[i],cutst[counter]);
        i++;counter++;
    }
    counter=0;
    while(counter<cuttcounter)
    {
        strcpy(member[i],cutt[counter]);
        counter++;i++;
    }
    /* prepare to return cutt cutst and cuttcounter and cutstcounter */
}
/* Warning ! : don't use s[i].d[0] it will be NULL */

```

ส่วนการเลือกเวลานัดที่แน่นอน

```

#include <ctype.h>
/* itos :function
return :string
do :itos by ctype isn't work thus I must write this function myself
*/
char *itos(i)
int i;
{
    char *c;

```

```

int o;

int counter=0;;

c=malloc(7);

o=i/10000;

if(o>0)

{
    c[counter]=o+'0';

    i=i-(10000*o);

    counter++;

}

o=i/1000;

if(o>0)

{

    c[counter]=o+'0';

    i=i-(1000*o);

    counter++;

}

o=i/100;

if(o>0)

{

    c[counter]=o+'0';

    i=i-(100*o);

    counter++;

}

o=i/10;

if(o>0)

{

    c[counter]=o+'0';

    i=i-(10*o);

    counter++;

}

o=i;

c[counter]=o+'0';

counter++;

c[counter]=0;

```

```

        return c;
    }

    char *nday(i)
    int i;
    {
        if(i==0)
            return "sun";

        if(i==1)
            return "mon";

        if(i==2)
            return "tue";

        if(i==3)
            return "wed";

        if(i==4)
            return "thu";

        if(i==5)
            return "fri";

        if(i==6)
            return "sat";

        return " ";
    }

    char *cl(b)
    int b;
    {
        int i,o;
        char *buf;
        char *buf2;
        char *rbuf;

        i= (b/4)+7;
        o= (b%4)*15;
        buf= malloc(10);
        buf2=malloc(10);
        rbuf=malloc(20);
        strcpy(buf,itos(i));

```

```

        if(o!=0)
            strcpy(buf2,itos(o));
        else
            strcpy(buf2,"00");
        i=0;
        while(buf[i]!='\0')
        {
            rbuf[i]=buf[i];
            i++;
        }
        rbuf[i]='\0';
        o=0;
        i++;
        while(buf2[o]!='\0')
        {
            rbuf[i]=buf2[o];
            i++;
            o++;
        }
        rbuf[i]='\0';
        return rbuf;
    )

```

```
char *convertallsc(allsc)
```

```
char *allsc[7];
```

```
{    int i,o;
```

```
    int d=10,b,e;
```

```
    int counter;
```

```
    char *buf;
```

```
    char *pbuf;
```

```
    for(i=0;i<7;i++)
```

```
        for(o=0;o<=60;o++)
```

```
        {
```

```
            if(allsc[i][o]==' '||allsc[i][o]=='O')
```

```

d=i;
b=0;
while(allsc[i][o]!='&&allsc[i][o]!='#')
{
    o++;
}
e=0;
o=62;i=8;
}

}

if(!(i>7&&o>61))
{
    return " ";
}
buf=malloc(30);
pbuf=malloc(10);
strcpy(buf,nday(d));
buf[3]='\0';
strcpy(pbuf,d(b));

i=0;
counter =4;
while(pbuf[i]!=0)
{
    buf[counter]=pbuf[i];
    i++;
    counter++;
}

buf[counter]='\0';
counter++;
strcpy(pbuf,d(e));
i=0;
while(pbuf[i]!=0)

```

```

        {
            buf[counter]=pbuf[i];

            i++;

            counter++;

        }

        buf[counter]=0;

        return buf;

    }

char *mapmember(member,t,tmax,s,smax,classtime,a,b,c)
char *member[20];
struct table t[MAX_TEACH];
int tmax;
struct table s[MAX_STD];
int smax;
char classtime[28][20];
int a;
int b;
int c;
{
    int pointt[MAX_TEACH],points[MAX_STD];
    char makex[MAX_MENU1];
    int i,o;
    int buff[MAX_MENU1];
    int buf;
    int psc;
    int counter;
    int counts;
    char *allsc[7];
    char *bufday1;
    int b1,e;
    bufday1=malloc(10);
    for(i=0;i<7;i++)
    {

        allsc[i]=malloc(62);

        for(o=0;o<62;o++)

```

```

        allsc[i][o]=' ';
        allsc[i][o]=0;
    }

    for(i=1;i<=member[0][0];i++)
    {
        points[i-1]= findsnumber(s,smax,member[i]);
        pointt[i-1]= findsnumber(t,tmax,member[i]);
    }

    /* for mapping of student */
    for(counts=0;counts<member[0][0];counts++)
    {
        psc=points[counts];
        if(psc!=1)
        {
            for(counter=1;counter<
under+++)
            {
                strcpy(buff,s[psc].d[counter]);
                strcpy(bufday1,search(buff,1));
                b1=genx(bufday1);
                e=genend(bufday1);
                o=day(search(buff,0));
                for(i=b1;i<e;i++)
                {
                    allsc[o][i]='#';
                }
            }
        }
    }

    /* for mapping of teacher schedule */
    for(counts=0;counts<member[0][0];counts++)

```

```

psc=pointt[counts];
if(psc!=1)
{
    for(counter=1;counter<t[psc].number;counter++)
    {
        strcpy(buff,t[psc].d[counter]);
        strcpy(bufday1,search(buff,1));
        b1=genx(bufday1);
        e=genend(bufday1);
        o=day(search(buff,0));
        for(i=b1;i<e;i++)
        {
            allsc[o][i]='#';
        }
    }
}
if(b>59)
    b=59;
if(c>6)
    c=6;
if(a>b)
{
    i=a; /* c mean day and a to b is time to time */
    l=b;
    b=i;
}
current_itemy=c;
current_itemx=b;
for(i=a;i<b;i++)
{
    if(allsc[c][i]!='#')
        allsc[c][i]='.';
    else

```

```

        allsc[c][i]='0';
    }

    mk(allsc);

    strcpy(makex,convertallsc(allsc));

    return makex;
}

day(buf)
char buf[3];
{
    char str[3];
    strcpy(str,tolower(buf));

    if(strcompare(str,"sun",3)==0)
        return 0;
    else if(strcompare(str,"mon",3)==0)
        return 1;
    else if(strcompare(str,"tue",3)==0)
        return 2;
    else if(strcompare(str,"wed",3)==0)
        return 3;
    else if(strcompare(str,"thu",3)==0)
        return 4;
    else if(strcompare(str,"fri",3)==0)
        return 5;
    else if(strcompare(str,"sat",3)==0)
        return 6;
    else return -1;
}

/*
makeupclass : function
act : make up menu
return: make up class time(make x)
*/

char *makeclass(member,t,tmax,s,smax)
char *member[20];

```

```

struct table t[MAX_TEACH];

int tmax;

struct table s[MAX_STD];

int smax;

{
    int i;

    int a,b,c;

    int checkmistake;

    int counter;

    char classtime[28][20]; /* each day have no more than 4 subject */

    char sb[40];

    char makex[MAX_MENU1];

    char buf[7];

        chelp(1,'D','');
        chelp(3,'A',"Add time");
        chelp(3,'M',"Accep it");
        chelp(0,'W',"Week make");
        msg_reverse(21,"put string day/ begintime / endtime [CTRL_A]");
        move(10,2);
        attrset(A_REVERSE);
        attron(A_REVERSE);
        addstr("Map time:");
        do{

            chelp(0,'W',"Week make");

            msg_reverse(21,"put string day/ begintime / endtime [CTRL_A]");

            clearpage(11,20);

            checkmistake=0;

            move(10,12);

            attroff(A_REVERSE);

            readlinemk(sb,40);

            if(sb[0]>26)

                {

                    for(i=0;j<=3;j++)

                        buf[j]=sb[j];

                    buf[3]=0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

checkmistake=day(buf);
c=checkmistake;
if(checkmistake!=1)
{
    current_itemy=checkmistake;
    counter=0;
    while(sb[i]!='&&counter<7)
    {
        buf[counter]=sb[i];
        counter++;
        i++;
    }
    i++;
    if(counter<7)
    {
        buf[counter]='-';
        counter++;
        while(sb[i]!=0&&counter<14)
        {
            buf[counter]=sb[i];
            counter++;
            i++;
        }
        buf[counter]=0;
        if(counter<14)
        {
            a=genx(buf);
            b=genend(buf);
            if(a<0||b<0)
            {
                checkmistake=-1;
                sb[0]=CTRL_A;
            }
            else checkmistake=-1;
        }
    }
}

```

```

    }

    }else
    {

        a=0;

        b=0;

        c=0;

    }

    if(sb[0]==CTRL_A&&checkmistake!=1)
    {

        chelp(0,'K','map time');
        move(11,12);
        stropy( makex,mapmember(member,t,tmax,s,smax,classtime,a,b,c));
        move(10,31);
        /* Point! if you make new window don't miss to change it*/
        addstr(makex);
        addstr"  ";
        refresh();
        checkexitmenu1=checkexitmenum;
        chelp(1,'K',' ');

    }

    else checkexitmenu1=sb[0];

    if(makex[0]==' '&&checkexitmenu1==CTRL_M)
    {

        beep();

        msg_reverse(21,"You must make any mark by CTRL_A before accept");
        checkexitmenu1=CTRL_A;

    }

    if(checkexitmenu1==CTRL_H)
    {

        hv("hmakeup");
        checkexitmenu1=CTRL_A;
    }

```

```

    }

    }

while(checkexitmenu1!=CTRL_X&&checkexitmenu1!=CTRL_E&&checkexitmenu1!=CTRL_F&&checkexitmenu1!
=CTRL_L&&checkexitmenu1!=CTRL_M&&checkexitmenu1!=CTRL_W);

    attroff(A_REVERSE);

    move(10,2);

    addstr("Map time:");

    refresh();

    return makex;

}

```

ส่วนเสริมของการนับเวลาที่แน่นอน

```

/* readlinemk : function
do      : extra work with makeup menu only
return  : type of finishing of input string
environment: curses
*/
readlinemk(st,limitchar)
char st[80];
char limitchar;
{
    char c,x,y,xbuf,ybuf;
    int i;
    int everb=0;
    int counter=0;
    addstr(" (day/hh:mm/hh:mm) ");
    refresh();
    getyx(stdscr,ybuf,xbuf);

    do{
        getyx(stdscr,y,x);

        if(x>=xbuf&&y==ybuf); /*maximum character is 40*/

```

```

c=readch10;

if(c>=CTRL_A&&c<=CTRL_H)

/*WARN:do not use ctrl m,it is '\n'*/
{

    st[0]=c;

    st[1]='\0';

    return 1;

}

if(x>(xbuf+limitchar))

    move(y,x-1);

if(!((x==xbuf&&c<27)||(extrakey==1)))

{

    addch(c);

}

else if(extrakey==1)

{

    if((c==67||c==KEY_LEFT))

        move(y,x+1);

    if((c==68||c==KEY_RIGHT)&&x>xbuf)

        move(y,x-1);

}

refresh();

}

}while(c!='\n'&&c!='\r');

attron(A_REVERSE);

for(j=xbuf;j<x;j++)

{

    move(ybuf,j);

    c=inch0;

    addch(c);

```

```

        if(c==' '&&everb==0)
        {
            everb=0;
        }else
        {
            st[counter]=c;
            counter++;
            everb=1;
        }
    }

    st[counter]=0;
    attroff(A_REVERSE);
    return 0;
}

#include <ctype.h>
#include <string.h>
#include <strings.h>
#define MAX_MENU1 200 /* max of array on menu */
int current_itemx=0; /* golbol reserve for this menu1.h */
int current_itemy=0;
int checkexitmenu;
/* select : function
return : string that selected
do : receive user in put character or cross to select menu
same : as menu1.h
*/
char *mk(xmap)
char *xmap[70];
{
    int hassubject[7];
    int c=0;
    int x,y,i;
    int xb,yb;

```

```

savetty(); /* save tty of system*/
cbreak();nonl();noecho();standend(); /* set use Ctrl-break
                                and no echo to getch */
 keypad(stdscr,TRUE); /* enable keypad to use extra keyboard*/
msg_reverse(21,["Press UP DOWN LEFT RIGHT key"]);
getyx(stdscr,y,x); /* from where */
for(i=0;i<7;i++)
{
    move(y+i,x);
    addstr(" ");
}
refresh();
makeborder(y,x);
move(y+8,x+5);
addstr(" Your cursor at time : ");
do{
    move(y,x);
    /* goto where */
    checkkeymk(c,xmap);
    /* automatic modifi current item x and y*/
    /* check current key to translate mean*/
    pumk(xmap);
    /*display menu*/
    refresh();
    c=readch10;
}while(c!='\n'&&c!='\r'&&!(c>=2&&c<=26)); /* exit value not CTRL_A*/
    resetty(); /*      return tty environment */

if(checkexitmenu1!=CTRL_M)
    oldcheckexitmenu1=checkexitmenu1;

checkexitmenum=c;
}

```

บทที่ 4

การทดลอง

ในบทนี้กล่าวถึงการทดลองใช้งานโปรแกรมและสภาพแวดล้อมที่ใช้ในการส่งจดหมายเพื่อหาสภาพที่ทำการส่งจดหมายที่ดีและทดสอบว่าโปรแกรมสามารถใช้งานได้จริง



```

$ cat .profile

# Set up the terminal:

eval `tset -s -Q -m `?:vt100'`

stty erase ^H kill ^U intr ^C eof ^D

stty hupcl ixon ixoff

tabs

# Set up the shell environment:

set -u

trap "echo 'logout\n\nI hope to see you soon.\nGood luck\n'" 0

# Set up the shell variables:

EDITOR=pico

export EDITOR

.pfinc

if mhpath unseen +inbox >/dev/null 2>&1
then

echo ""

echo "===== You have UNREAD mail ====="

echo ""

scan +inbox unseen

echo "===== "

echo ""

fi

$

```

รูปแสดงการทำงานขณะผู้ใช้เข้าสู่ระบบ

จากรูป แสดงการทำงานขณะผู้ใช้เข้าสู่ระบบ นี้จะเห็นได้ว่าการติดตั้งค่าเอสทีทีวาย (stty) ของเทอร์มินัลในโครงงานนี้ใช้เทอร์มินัลแบบวีทีทีหนึ่งร้อย (vt100) จึงติดตั้งขณะเข้าระบบให้เป็นเทอร์มินัลวีทีทีหนึ่งร้อยตั้งเเต่แรกไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่แรกและในไฟล์นี้ภายในคำสั่งที่ติดตั้งเทอร์มินัลยังมีที่เซต(tset) เพื่อสามารถกำหนดเทอร์มินัลโดยผู้ใช้เองจะเข้าสู่ระบบได้ด้วยโดยถือค่ากำหนดตัวที่หนึ่งร้อย เป็นค่าพื้นฐานถ้าผู้ใช้ไม่กำหนดให้เป็นอย่างอื่นระบบก็จะกำหนดให้เทอร์มินัลนั้นเป็นเทอร์มินัลแบบวีทีหนึ่งร้อยไปโดยปริยาย จากนั้นก็จะเป็นการกำหนดค่าของตัวอักษรพิเศษต่างเช่น การลบอักษรข้างหน้าหนึ่งตัวเรียกคิวล์ (kill) กำหนดให้ต้องกดคีย์คอนโทรล เอช(Ctrl-H) เป็นต้น จากนั้นก็มีการกำหนดว่าถ้าผู้ใช้ออกจากระบบให้แสดงข้อความอะไรหลังจากที่ผู้ใช้ผู้นั้นออกจากระบบไปแล้ว ในส่วนต่อมาก็คือเป็นการ ติดตั้งสภาพแวดล้อมในการใช้งานโดยติดตั้งโปรแกรมอีดิเตอร์ (editor) ที่กำหนดว่าถ้ามีการเรียกใช้งานอีดิเตอร์เมื่อใดก็จะใช้โปรแกรมที่กำหนดไว้นี้เสมอ เช่น ในอาร์ทีน (rtin) เมื่อเราจะคุยกับใครหรือกลุ่มใดกลุ่มหนึ่งจะมีการเรียกใช้อิดิเตอร์ขึ้นมาในโครงงานนี้กำหนดให้อิดิเตอร์เป็นโปรแกรมชื่อพิโค (pico) ซึ่งเป็นอีดิเตอร์ที่ใช้กันอย่างแพร่หลายในปัจจุบัน และสามารถใช้งานได้งานเนื่องจากมีความคล้ายคลึงกันอีดิเตอร์อื่นที่มีในเครื่องพีซีที่ผู้ใช้ส่วนใหญ่คุ้นเคย โดยมีคำสั่งดังต่อไปนี้

```
# Set up the shell variables:
```

```
EDITOR=pico
```

```
export EDITOR
```

รูปแสดงการกำหนดสภาพแวดล้อมของเชลล์

อธิบายรูป แสดงการกำหนดสภาพแวดล้อมของเชลล์ โดยบรรทัดแรกมีเครื่องหมาย '#' นำหน้าแสดงว่าเป็นประโยคที่ไม่มีการปฏิบัติการป็นเพียงคำอธิบายของส่วนที่ตามมาเท่านั้น บรรทัดที่สองเป็นการกำหนดว่าโปรแกรมสำหรับเป็นอีดิเตอร์ถูกกำหนดให้เป็นโปรแกรมพิโค และบรรทัดที่สามเป็นคำสั่งที่นำข้อความในบรรทัดที่สองไปใช้งานจริงถ้าขาดบรรทัดที่สามนี้ข้อความในบรรทัดที่สองจะไม่ถูกนำไปปฏิบัติ

ส่วนต่อมากจะเข้าไปทำงานในส่วนของโปรแกรมโครงงาน โดยโปรแกรม .pfinc นี้จะทำการตรวจสอบหาคำหมายข้อความฉบับใหม่ว่ามีจดหมายข้อความใดเข้ามาบ้างหลังจากผู้ใช้ออกจากระบบครั้งสุดท้าย ถ้ามีจะนำจดหมายข้อความเหล่านั้นไปใส่ในลำดับของจดหมายข้อความใหม่และนำจดหมายข้อความที่อยู่ในลำดับของจดหมายข้อความที่ยังไม่เคยถูกเปิดอ่านมาไว้ในลำดับของจดหมายข้อความที่ผู้ใช้ยังไม่เคยเห็น แล้วทำการรวมจดหมายข้อความทั้งหมดเหล่านี้เข้าด้วยกัน แล้วทำการตรวจสอบว่ามีจดหมายข้อความที่ยังไม่เคยถูกเปิดอ่านอยู่ในตู้จดหมายของผู้ใช้นั้นหรือไม่ ถ้ามีจดหมายที่ยังไม่เคยถูกเปิดอ่านเลยในตู้จดหมายของผู้ใช้ผู้นั้นแล้วก็จะแสดงหัวของจดหมายข้อความนั้นออกมาว่าใครเป็นผู้ส่งมา ส่งมาเมื่อใด และหัวเรื่องของจดหมายข้อความฉบับที่ผู้ใช้ไม่เคยอ่านนั้นมีที่ฉบับอะไรบ้าง

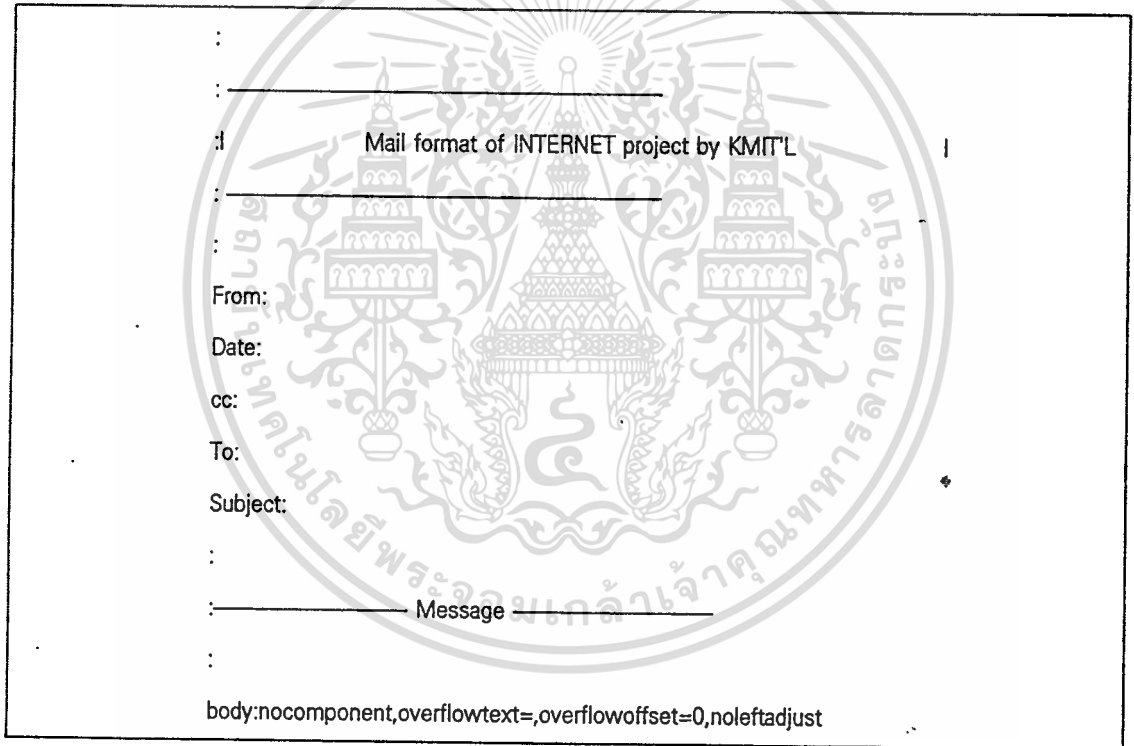
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Path: Mail
 Unseen-sequence: unseen
 Folder-protect: 700
 Msg-protect: 600
 Signature: "Chaitad Huangsamphan"
 repl: -noannotate
 scan: -form scan.mailx
 show: -form mhl.t
 show: -showproc mhl
 showproc: mhl

รูปแสดงค่าที่กำหนดในระบบจัดการจดหมาย

ในไฟล์โปรไฟล์ของระบบจัดการจดหมายข้อความนี้ จะเป็นตัวกำหนดให้กล่องรับจดหมายข้อความอยู่ในไดเรกทอรีใดในที่นี้จะมีคำสั่ง : Path: Mail ซึ่งระบุว่าให้กล่องรับจดหมายข้อความนั้นอยู่ในไดเรกทอรี Mail และในบรรทัดถัดมาก็จะกำหนดว่ามีชุดจดหมายใดบ้างชุดจดหมายนี้อยู่ที่ใดและชุดจดหมายเหล่านี้ชื่ออะไรบ้างการตั้งชื่อของชุดจดหมายเหล่านี้ควรจะตั้งให้สอดคล้องกับงานจริงที่ใช้กับชุดจดหมายต่างๆเหล่านั้น ในที่นี้ชุดจดหมายที่สร้างขึ้นมานี้ทำงานกับจดหมายที่ผู้เป็นเจ้าของไม่เคยอ่านมาก่อนจึงให้ชื่อดังเช่นที่ปรากฏนี้ คำสั่งที่ระบุชุดกำหนดคือ Unseen-sequence: unseen จากนั้นขอให้ท่านพิจารณาบรรทัดต่อมาในรูป **แสดงค่าที่กำหนดในระบบจัดการจดหมาย** จะเห็นคำสั่งที่ใช้ในการให้สิทธิในการใช้กล่องรับจดหมายร่วมกันกับเราในโคตรงานที่กำหนดนี้จะระบุกำหนดไว้เป็น 700 ซึ่งเมื่อเทียบเป็นไบนารีของแต่ละหลักแล้วจะได้เป็น 111 000 000 โดยที่แต่ละหลักของไบนารีมีความหมายเป็นจริง หรือเท็จ ถ้าไบนารีหลักใดเป็น 1 ก็จะแสดงว่าให้สิทธิแก่ผู้ใช้ที่ระบุในหลักของไบนารีในหลักนั้นให้สามารถทำงานกับกล่องรับจดหมายข้อความได้ โดยไบนารีสามหลักแรกจะหมายถึงเจ้าของกล่องจดหมายข้อความนั่นเอง ไบนารีสามหลักต่อมาหมายถึงกลุ่มบุคคลที่เป็นกลุ่มที่เจ้าของเป็นสมาชิกอยู่ และไบนารีสามหลักต่อมาจะหมายถึงการให้หรือไม่ให้สิทธิแก่บุคคลทั่วไป โดยหลักแรกของกลุ่มจะหมายถึงให้สิทธิในการอ่านแก่บุคคลในกลุ่มนั้นหรือไม่ ส่วนหลักที่สองของกลุ่มคือระบุว่าให้สิทธิในการเขียนลงในกล่องจดหมายข้อความนั้นหรือไม่ และหลักที่สามของกลุ่มคือระบุว่าให้สิทธิในการทำงานกับกล่องจดหมายแก่กลุ่มนั้นหรือไม่ ท่านคงเห็นว่าในโคตรงานกำหนดไว้เป็น 700 แสดงว่าให้สิทธิเฉพาะแก่เจ้าของกล่องจดหมายข้อความเท่านั้นในการอ่าน เขียน และทำงานกับกล่องจดหมายข้อความของตน และไม่ให้สิทธิแก่ผู้ใดในกลุ่มหรือบุคคลทั่วไปในการอ่าน เขียน หรือ ทำงานแก่กล่องรับจดหมายข้อความของเจ้าของเลย จากนั้นก็จะทำการป้องกันตัวจดหมายข้อความเองว่าจดหมายข้อความจะให้บุคคลในกลุ่มหรือบุคคลทั่วไปสามารถอ่าน หรือเขียน จดหมายข้อความเหล่านั้นได้หรือไม่ จากนั้นก็จัดการกับการแสดงจดหมายข้อความโดยให้มีการแสดงเป็นแบบเดียวกันกับการแสดงจดหมายข้อความในเมลเอกซ์ (mailx) ซึ่งจะแสดงเลขที่ของจดหมายข้อความเป็นอันดับแรกจากนั้นก็แสดงชื่อของผู้ส่งจดหมายข้อความนั้น จากนั้นก็จะแสดงวันเดือนปี โดยการแสดงจะแสดงเดือนเป็นตัวอักษรย่อของเดือนนั้นซึ่งจะทำให้ง่ายต่อการมองว่าตัวเลขหลักใดเป็นวันหลักใดเป็นเดือนหลักใดเป็นปีขอให้ท่านเทียบไม่วารณใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งหากท่านจะใช้

ของเดือนนั้นซึ่งจะทำให้ง่ายต่อการมองว่าตัวเลขหลักใดเป็นวันหลักใดเป็นเดือนหลักใดเป็นปีขอให้ท่านเทียบ
การแสดงวันที่เป็นแบบตัวเลขทั้งหมดซึ่งเป็นการแสดงจดหมายข้อความแบบมาตรฐานกับการแสดงวันที่ที่ได้ผ่าน
การปรับปรุงในโครงการนี้ ตัวอย่างเช่น แสดงวันที่ 10 ธันวาคม 1994 การแสดงแบบตัวเลขเป็นดังนี้ 10/12/94
เทียบกับการแสดงวัน-เดือน-ปีในโครงการแล้ว 10 Dec 94 จะเห็นได้ว่าไม่เกิดความสับสนระหว่างว่าเป็นวันที่ 10
เดือนธันวาคม หรือ วันที่ 12 เดือนตุลาคม จากนั้นก็ทำการกำหนดรูปแบบการแสดงผลจดหมายข้อความ โดยมีการ
กำหนดรูปแบบไว้แล้วล่วงหน้าในไฟล์รูปแบบซึ่งมีชื่อว่า เอ็มเอชแอล.ที เนื่องจากส่วนมากแล้วหัวจดหมายข้อ
ความจะมีความยาวมากและทำให้ผู้ใช้สับสนว่าส่วนใดเป็นหัวเนื้อจดหมายจริงๆ หรือจะเริ่มอ่านจากส่วนใดการ
ทำการจัดการนี้จะเป็นการช่วยลดหัวจดหมายที่ผู้ใช้ไม่จำเป็นต้องทราบออกไป เช่นการส่งจดหมายนั้นส่งผ่านเส้น
ทางใดมา



:
 :
 : Mail format of INTERNET project by KMITL :
 :
 :
 From:
 Date:
 cc:
 To:
 Subject:
 :
 : Message :
 :
 body:nocomponent,overflowtext=,overflowoffset=0,noleftadjust

รูปแสดงไฟล์ที่ใช้จัดรูปแบบในการแสดงจดหมายข้อความ

บทที่ 5

สรุป

ในบทนี้กล่าวถึงผลของการทดสอบในบทที่ 4 และกล่าวถึงประโยชน์ที่อำนวยความสะดวกผู้ใช้โปรแกรม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประโยชน์แก่ผู้ใช้

1. ผู้ใช้สามารถประกาศการเลื่อนตารางเรียนของรายวิชาต่างๆในสัปดาห์ที่ต้องการโดยไม่ต้องจดจำผู้เข้าเรียนหรือผู้สอนเพียงแต่ทราบรายวิชาก็สามารถส่งจดหมายเป็นประกาศแก่ผู้เกี่ยวข้องได้
2. ผู้ใช้สามารถประกาศการนัดตารางเรียนของรายวิชาต่างๆในสัปดาห์ที่ต้องการโดยไม่ต้องจดจำผู้เข้าเรียนหรือเวลาว่างของผู้เกี่ยวข้องในวิชาเรียนโดยโปรแกรมจะเป็นผู้แสดงให้เห็นถึงเวลาที่ว่างร่วมกันเพื่อทำการนัดเวลาเรียน
3. นักศึกษาสามารถส่งการบ้านผ่านเครือข่ายอินเทอร์เน็ต(internet)โดยเลือกเพิ่มข้อมูลที่ทำการส่งจากเมนู
4. ผู้ใช้สามารถอ่านข่าวสารจากบอร์ดข่าวสารที่แยกประเภทของข่าวสารตามหัวข้อ
 - 4.1 ข่าวการเลื่อนเวลาของวิชาเรียน
 - 4.2 ข่าวการนัดเวลาเรียนพิเศษ
 - 4.3 ผลของการบ้านที่อาจารย์ตรวจเสร็จ
 - 4.4 ข่าวประกาศทั่วไป
5. ผู้ใช้สามารถส่งจดหมายไปหาอาจารย์ได้โดยเลือกรายชื่อของอาจารย์จากเมนู
6. อาจารย์สามารถใช้การป้อนคำสั่ง(command line)เพื่อตรวจการบ้านได้มีประสิทธิภาพกว่าระบบรับจดหมายแบบเมนู
7. ถ้าผู้ใช้มีความสงสัยการทำงานสามารถศึกษาโปรแกรมได้จากระบบช่วยเหลือ(help)
8. เพิ่มความสะดวกในการทำงานด้วยระบบแนะนำขั้นตอนการทำงาน
9. การเปลี่ยนแปลงแต่ละโปรแกรมในข้อที่ผ่านมาสามารถตั้งตำแหน่งของโปรแกรมได้ด้วย setup และถ้าโปรแกรมไม่สามารถทำงานได้โดยปกติโปรแกรม menu ซึ่งเป็นโปรแกรมบริหารโปรแกรมอื่นๆจะดูแลไม่ให้ระบบหยุดชะงัก(hang)

สรุป

ในการทำโครงงานในครั้งนี้ เป็นการทำให้โปรแกรมเพื่อการศึกษาโดยโปรแกรมนี้จะทำงานร่วมกับระบบอินเทอร์เน็ตที่มีระบบปฏิบัติการยูนิกซ์ โดยมีการออกแบบและสร้างโปรแกรมที่ใช้กับงานจดหมายข้อความตอบรับโดยในการอ่านจดหมายข้อความแต่ละฉบับนั้นจะมีการแยกประเภทของจดหมายไว้เป็นชนิดก่อน เช่นมีการแยกจดหมายข้อความที่เกี่ยวข้องกับการนัดเลื่อนชั้นเรียนออกจากจดหมายข้อความที่ใช้เป็นสื่องานประกาศทั่วๆไป เพื่อให้งานต่อการใช้งานแยกความแตกต่างระหว่างจดหมายข้อความต่างๆอย่างเด่นชัด เมื่อท่านเข้าสู่โปรแกรมสามารถจะเลือกดูเฉพาะกล่องจดหมายข้อความที่ท่านสนใจได้อย่างชัดเจน

นอกจากนั้นยังได้แก้ปัญหาในเรื่องที่ท่านส่งจดหมายไปแล้วแต่ไม่ได้รับการตอบกลับว่าผู้ที่ท่านส่งถึงนั้นได้รับจดหมายข้อความที่ท่านส่งไปให้หรือไม่ ในโครงงานนี้ได้แก้ปัญหาในเรื่องนี้โดยเมื่อจดหมายข้อความนั้นปรากฏต่อสายตาผู้รับเป็นครั้งแรก โดยผู้รับนั้นไม่ต้องทำการตอบจดหมายด้วยตัวเองทุกฉบับ โปรแกรมจะทำการตอบจดหมายที่ปรากฏเป็นครั้งแรกให้ทุกฉบับทันทีเหมือนเป็นการส่งจดหมายลงทะเลเป็น

ในโปรแกรมโครงงานจะแก้ปัญหาการที่ผู้ใช้ไม่ได้เข้าไปอ่านจดหมายข้อความในกล่องรับจดหมายข้อความของตนโดยการจะแสดงจดหมายใหม่ที่ได้รับเข้ามารวมถึงจดหมายที่ได้รับเข้ามาแล้วแต่ยังไม่ได้เปิดอ่านโดยการแสดงจดหมายข้อความดังกล่าวจะช่วยให้ท่านเห็นจดหมายจากผู้ส่งที่ท่านคาดหวัง หรือรอคอยอยู่ได้ และยังหาหัวเรื่องที่ท่านสนใจขณะที่ท่านทำการเข้าสู่ระบบได้อีกด้วย จึงเป็นการสะดวกอย่างยิ่งในการที่มีการแสดงจดหมายข้อความที่ท่านยังไม่ได้อ่านขณะที่ผู้ใช้เข้าสู่ระบบ

มีการจัดส่งไฟล์การบ้าน โดยการส่งไฟล์การบ้านจะทำการรวมไฟล์นั้นรวมกับจดหมายข้อความนั้นด้วยและเมื่อจะตรวจการบ้าน ก็จะแยกไฟล์เหล่านั้นออกจากจดหมายข้อความได้ด้วย แม้แต่การรวมไฟล์เข้าในจดหมายข้อความธรรมดาท่านก็สามารถทำได้ ทำให้สามารถส่งไฟล์ที่สามารถปฏิบัติการได้เลยโดยที่ผู้รับไม่ต้องนำไฟล์ไปคอมไพล์ใหม่แต่อย่างใด

กิตติกรรมประกาศ

ในการทำงานในโครงการนั้นนอกจากจะได้รับความกรุณาจาก บิดา-มารดา, อาจารย์ที่ปรึกษา และ อาจารย์ทุกท่านในคณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง แล้วยังได้รับความช่วยเหลือเป็นอย่างดีจากท่านต่าง ๆ ดังต่อไปนี้

- | | |
|------------------------------|--|
| 1. ดร. สมนึก ศิริโต | อาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ ม.เกษตร ให้คำแนะนำ |
| 2. อ. สุรศักดิ์ สงวนพงศ์ | อาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ ม.เกษตร ให้คำแนะนำ |
| 3. อ. สมชาย นำประเสริฐชัย | อาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ ม.เกษตร ให้คำแนะนำ |
| 4. คุณวรรณฤดี ลาภไพบุลย์พงศ์ | ช่วยหาและแนะนำหนังสือที่ใช้ในการทำโครงการนี้ |
| 5. พี่ สรวิศ บุญมี | ช่วยหาโปรแกรมและแนะนำวิธีติดตั้งโปรแกรมที่ใช้ในโครงการ |
| 6. เพื่อนๆทุกคน | ที่อยู่เป็นช่วยทำงานและให้กำลังใจ |

สุดท้ายกระผมทั้งสองในนามของผู้จัดทำโครงการบริการเพิ่มเติมด้านการศึกษาในระบบอินเทอร์เน็ต จึงใคร่ขอกราบขอบพระคุณทุกท่านที่ให้ความกรุณาช่วยเหลือการทำงานในโครงการนี้มาใน โอกาสนี้ด้วย

หนังสืออ้างอิง

1. Bryan Costales with Eric Allman & Neil Rickert, "Help for UNIX System Administrators", O'Reilly & Associates, Inc., 670p., 1993.
2. John Strang, "Programming with curses", O'Reilly & Associates, Inc., 71p., 1986.
3. Douglas E. Comer and David L. Stevens, "Internetworking With TCP/IP", Prentice-Hall International, Inc., 441p., 1987.
4. Lowell Jay Arthur, "UNIX Shell Programming", John Wiley & Sons, Inc., 438p, 1990.
5. สมพงษ์ บุญธรรมจินดา, "คู่มือติดตั้งและใช้งานโมเด็ม/แฟกซ์โมเด็ม ด้วยตัวเอง", บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน), 158 หน้า, 2521
6. ชรินทร์ เขาวมิต, "การโปรแกรมมิ่ง ยูนิกซ์ เซลล์", บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน), 275.หน้า, 2521

