

ระบบสนับสนุนการค้นคืนบทความวิทยานิพนธ์ผ่านเครือข่าย



นาย โสภณ ทองเกษม



สถาบันวิทยบริการ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

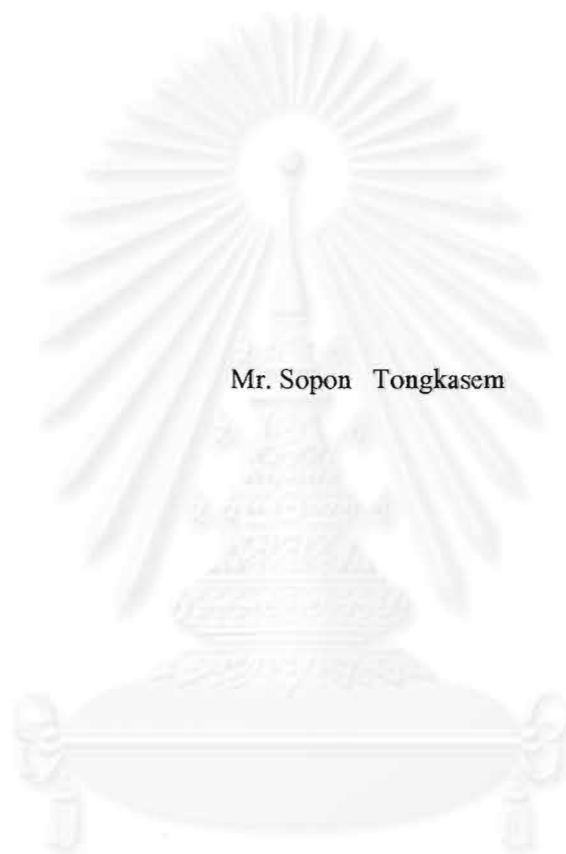
บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2542

ลิขสิทธิ์ของบัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย

ISBN 974-333-032-1

THESIS ABSTRACT RETRIEVAL SUPPORT SYSTEM VIA A NETWORK



Mr. Sapon Tongkasem

สถาบันวิทยบริการ

จุฬาลงกรณ์มหาวิทยาลัย
A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science

Department of Computer Engineering

Graduate School

Chulalongkorn University

Academic Year 1999

ISBN 974-333-032-1

หัวข้อวิทยานิพนธ์ ระบบสนับสนุนการค้นคืนบทความด้วยวิทยานิพนธ์ผ่านเครือข่าย
โดย นาย โสภณ ทองเกษม
ภาควิชา วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร. สมชาย ประสิทธิ์จตุระกุล

บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้บัณฑิตวิทยาลัยเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาโทบัณฑิต

วิมล วิมล

..... คณบดีบัณฑิตวิทยาลัย

(รองศาสตราจารย์ ดร. สุชาดา กิระนันท์)

คณะกรรมการสอบวิทยานิพนธ์

Prasert Jongsitthirattana

..... ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ กอบกุล เตชะวณิช)

วิมล วิมล

..... อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร. สมชาย ประสิทธิ์จตุระกุล)

วิมล วิมล

..... กรรมการ

(อาจารย์ ดร. จิต ศิริบูรณ์)

ประภาส จงสิตยวิวัฒนา

..... กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร. ประภาส จงสิตยวิวัฒนา)

โศภณ ทองเกษม : ระบบสนับสนุนการค้นคืนบทคัดย่อวิทยานิพนธ์ผ่านเครือข่าย
(THESIS ABSTRACT RETRIEVAL SUPPORT SYSTEM VIA A NETWORK)

อาจารย์ที่ปรึกษา : ผศ. ดร. สมชาย ประสิทธิ์จตุระกุล , 140 หน้า. ISBN 974-333-032-1.

วิทยานิพนธ์นี้นำเสนอการออกแบบและพัฒนาระบบสนับสนุนการค้นคืนเอกสารไทยบน
เว็ลต์ไวด์เว็บ โดยใช้ข้อมูลบทคัดย่อวิทยานิพนธ์ของจุฬาลงกรณ์มหาวิทยาลัยเป็นกรณีศึกษา ระบบนี้
ประกอบด้วยส่วนจัดเตรียมแฟ้มข้อมูล ส่วนติดต่อกับผู้ใช้เพื่อรับข้อความ ส่วนประมวลผลข้อความ
สำหรับค้นคืน และส่วนแสดงผลการค้นคืน แฟ้มข้อมูลขาเข้าที่ใช้เป็นแฟ้มข้อมูลแบบอธิบายตนเองที่
ถูกแปลงรูปแบบใหม่ให้เป็นแฟ้มสำหรับการจัดทำดัชนีและแฟ้มสำหรับการแสดงผลในรูปแบบเซชที่
เอ็มแอล ขั้นตอนการแปลงนี้อาศัยแฟ้มต้นแบบที่เก็บลักษณะของเซชข้อมูลและลักษณะของแฟ้มแสดง
ผลที่ต้องการได้จากการแปลง ในส่วนการจัดเตรียมแฟ้มข้อมูลและส่วนประมวลผลข้อความนั้นใช้ขั้น
ตอนวิธีจัดดัชนีข้อความไทยที่แยกข้อความไทยออกเป็นรายการคำ จากการทดสอบพบว่าระบบนี้
ต้องการเนื้อที่หน่วยความจำสำรองเพิ่มเติมเป็นขนาดประมาณสี่เท่าของขนาดของแฟ้มข้อมูลขาเข้า
และมีความสัมพันธ์ของเวลาการประมวลผลต่อคำร้องแปรผันเป็นเชิงเส้นกับขนาดของแฟ้มข้อมูลด้วย
ความลาดชันโดยเฉลี่ยต่ำที่ประมาณ 0.07 วินาทีต่อเมกะไบต์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา วิศวกรรมคอมพิวเตอร์
สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์
ปีการศึกษา 2542

ลายมือชื่อนิติต
ลายมือชื่ออาจารย์ที่ปรึกษา ผศ.ดร. ประสิทธิ์จตุระกุล
ลายมือชื่ออาจารย์ที่ปรึกษาร่วม

3972306621 : MAJOR COMPUTER SCIENCE
KEY WORD: WWW/HTML/CGI/THESIS ABSTRACT/TEXT RETRIEVAL
SOPON TONGKASEM : THESIS ABSTRACT RETRIEVAL SUPPORT SYSTEM VIA A NETWORK. THESIS ADVISOR : ASST. PROF. SOMCHAI PRASITJUTRAKUL, Ph.D. 140 pp. ISBN 974-333-032-1.

This thesis presents a design and development of a Thai-text retrieval support system via World Wide Web using Thesis Abstracts of Chulalongkorn University as a case study. The system consists of four major parts, data preparation, query-term user interface, query processing, and query result presentation. The input file is a self-describing structured file which is transformed into a set of files used for indexing and another set of files used for presentation, both are in HTML format. This is done using template files describing data field attributes and skeleton of the output files. A Thai-text indexing algorithm is used during data preparation and query processing for separating a given text into a list of words. Experimental results showed that the system requires four times (of the input file size) more storage space for indices, and the query time is approximately a linear function of the input file size with a small slope of 0.07 second per megabytes.

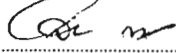


สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา.....วิศวกรรมคอมพิวเตอร์.....

สาขาวิชา.....วิทยาศาสตร์คอมพิวเตอร์.....

ปีการศึกษา..... 2542

ลายมือชื่อนิสิต..... 

ลายมือชื่ออาจารย์ที่ปรึกษา..... 

ลายมือชื่ออาจารย์ที่ปรึกษาร่วม.....



กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ ด้วยความช่วยเหลืออย่างดียิ่งของผู้ช่วยศาสตราจารย์ ดร. สมชาย ประสิทธิ์จิตรระกุล อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งท่านได้ให้คำแนะนำและข้อคิดเห็นต่างๆ ในการวิจัยมาด้วยดีตลอด

ขอขอบคุณท่านอาจารย์และเจ้าหน้าที่จากภาควิชาวิศวกรรมคอมพิวเตอร์ เจ้าหน้าที่จากหอสมุดกลาง ที่ได้ให้คำแนะนำและช่วยเหลือ ตลอดจน พี่ เพื่อนและน้องๆ ทุกคนจากบริษัท กฤษดา มหานคร จำกัด (มหาชน) ที่คอยช่วยเหลือและให้กำลังใจมาโดยตลอด

ทำยนี้ ผู้วิจัยใคร่ขอกราบขอบพระคุณ บิดา-มารดา นำผู้มีพระคุณ และบุคคลในครอบครัวของผู้วิจัยซึ่งคอยให้กำลังใจแก่ผู้วิจัยเสมอจนสำเร็จการศึกษา

โสภณ ทองเกษม

กันยายน 2542

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

| | หน้า |
|--|------|
| บทคัดย่อภาษาไทย | ง |
| บทคัดย่อภาษาอังกฤษ | จ |
| กิตติกรรมประกาศ | ฉ |
| สารบัญ | ช |
| สารบัญตาราง | ฅ |
| สารบัญภาพ | ญ |
| บทที่ | |
| 1. บทนำ | |
| ความเป็นมาและปัญหา | 1 |
| วัตถุประสงค์ของการวิจัย | 2 |
| ขอบเขตการวิจัย | 2 |
| ขั้นตอนการวิจัย | 2 |
| ประโยชน์ที่คาดว่าจะได้รับ | 3 |
| 2. แนวคิดและทฤษฎีที่เกี่ยวข้อง | 4 |
| โครงสร้างแฟ้มข้อมูล | 4 |
| ขั้นตอนวิธีที่ใช้จัดทำดัชนีคำ | 6 |
| เว็ลด์ไวด์เว็บและการสื่อสาร | 6 |
| 3. การวิเคราะห์และออกแบบระบบ | 16 |
| การจัดเตรียมแฟ้มดัชนีและแฟ้มสำหรับแสดง | 17 |
| การรับข้อความคำถามจากผู้ใช้เพื่อค้นหา | 19 |
| การปรับปรุงข้อความคำถามที่รับจากผู้ใช้ก่อนที่จะส่งไปโปรแกรมค้นหา | 19 |
| การปรับปรุงผลการค้นหาที่ได้รับจากโปรแกรมค้นหา | 20 |
| 4. การพัฒนาระบบสำหรับจัดเตรียมแฟ้มข้อมูล | 21 |
| การพัฒนาโปรแกรมจัดทำดัชนีคำ | 21 |
| การพัฒนาโปรแกรมจัดเตรียมแฟ้มข้อมูล | 25 |

สารบัญ (ต่อ)

| | หน้า |
|--|------|
| 5. การพัฒนาระบบสำหรับการค้นคืนบทความวิทยานิพนธ์ | 35 |
| การพัฒนาส่วนจัดเตรียมข้อมูลแบบควบคุมระยะไกล | 36 |
| การพัฒนาส่วนติดต่อกับผู้ใช้เพื่อรับข้อคำถาม | 40 |
| การพัฒนาส่วนค้นหาและแสดงผล | 41 |
| การติดตั้งระบบสำหรับเครื่องให้บริการ | 44 |
| 6. การวิเคราะห์ผลการทำงานและข้อเสนอแนะ | 45 |
| ทดสอบการจัดเตรียมข้อมูลสำหรับค้นคืน | 45 |
| ทดสอบการค้นคืนและแสดงผลลัพธ์ | 46 |
| ทดสอบการปรับปรุงโปรแกรมค้นหา SWISH-E | 54 |
| ข้อเสนอแนะ | 57 |
| รายการอ้างอิง | 58 |
| ภาคผนวก | |
| ภาคผนวก ก โปรแกรมค้นหา SWISH-E | 59 |
| ภาคผนวก ข โปรแกรมทดสอบระบบ WebBench | 63 |
| ภาคผนวก ค ตัวอย่างรายงานบทความวิทยานิพนธ์ | 66 |
| ภาคผนวก ง แสดงข้อคำถามที่ใช้ทดสอบการค้นคืนตามบทที่ 6 | 69 |
| ภาคผนวก จ ตัวอย่างการค้นคืนแบบต่างๆ | 71 |
| ภาคผนวก ฉ รายละเอียดแฟ้มรหัสต้นฉบับ | 74 |
| โปรแกรมจัดเตรียมแฟ้มข้อมูล | 77 |
| gnconfig.h | 77 |
| gnmem.h | 78 |
| gnstring.h | 78 |
| gnfile.c | 79 |
| gnmain.c | 80 |
| gnmem.c | 89 |
| gnread.c | 89 |

สารบัญ (ต่อ)

| | หน้า |
|---------------------------|------|
| gnstring.c | 100 |
| thaiword.c | 104 |
| makefile | 117 |
| โปรแกรมแบ่งคำ | 117 |
| tsepword.mak | 117 |
| ส่วนติดต่อกับผู้ใช้ | 118 |
| admin.htm | 118 |
| admin_h.htm | 118 |
| index.htm | 118 |
| search.htm | 119 |
| s_help.htm | 119 |
| s_main.htm | 120 |
| login.htm | 122 |
| menu.htm | 123 |
| โปรแกรมบทคำสั่งซีไอ | 124 |
| search.pl | 124 |
| admin.pl | 127 |
| mylib.pll | 137 |
| ประวัติผู้เขียน | 140 |

สารบัญตาราง

| | หน้า |
|---|------|
| ตารางที่ 2.1 แสดงรายละเอียดของส่วนหัวเขตข้อมูลของคำร้องขอ/คำตอบรับ | 10 |
| ตารางที่ 6.1 แสดงเนื้อหาที่ใช้ในการจัดเตรียมแฟ้มข้อมูล | 46 |
| ตารางที่ 6.2 แสดงการจัดเตรียมข้อความสำหรับการทดสอบ | 48 |
| ตารางที่ 6.3 แสดงรายละเอียดเงื่อนไขที่ระบุในชุดทดสอบของโปรแกรม WebBench | 49 |
| ตารางที่ 6.4 แสดงผลการทดสอบสำหรับแฟ้มแบบที่ 1 | 50 |
| ตารางที่ 6.5 แสดงผลการทดสอบสำหรับแฟ้มแบบที่ 2 | 51 |
| ตารางที่ 6.6 แสดงผลการทดสอบสำหรับแฟ้มแบบที่ 3 | 51 |
| ตารางที่ 6.7 แสดงผลการทดสอบสำหรับแฟ้มแบบที่ 4 | 52 |
| ตารางที่ 6.8 แสดงผลการทดสอบสำหรับแฟ้มแบบที่ 5 | 52 |
| ตารางที่ 6.9 แสดงผลการทดสอบสำหรับแฟ้มแบบที่ 6 | 53 |
| ตารางที่ 6.10 แสดงค่าเฉลี่ยเวลาที่ใช้ในการค้นคืนต่อคำร้อง | 53 |

สารบัญภาพ

| | หน้า |
|---|------|
| รูปที่ 3.1 แสดงการทำงานโดยรวมของระบบ | 17 |
| รูปที่ 3.2 แสดงการจัดเตรียมเพิ่มดัชนีและเพิ่มสำหรับแสดง | 18 |
| รูปที่ 4.1 แสดงรูปแบบการจัดเตรียมเพิ่มข้อมูล | 25 |
| รูปที่ 4.2 แสดงรูปแบบการจัดเตรียมเพิ่มข้อมูลผ่านเพิ่มเติมแบบ | 26 |
| รูปที่ 4.3 แสดงสิ่งที่เกี่ยวข้องกับโปรแกรมจัดเตรียมเพิ่มข้อมูล | 26 |
| รูปที่ 4.4 แสดงขั้นตอนการทำงานของโปรแกรมจัดเตรียมเพิ่มข้อมูล | 31 |
| รูปที่ 5.1 แสดงการเชื่อมต่อของการจัดเตรียมข้อมูลแบบควบคุมระยะไกล | 36 |
| รูปที่ 5.2 แสดงหน้าจอส่วนที่รับข้อความที่ต้องการค้นหาจากผู้ใช้ | 41 |
| รูปที่ 5.3 แสดงขั้นตอนการทำงานของโปรแกรม Search.pl | 42 |
| รูปที่ 5.4 ตัวอย่าง เว็บไซต์ที่แสดงผลการค้นหา | 43 |
| รูปที่ 5.5 ตัวอย่าง เว็บไซต์แสดงรายงานบทคัดย่อ | 43 |
| รูปที่ 5.6 แสดงโครงสร้างของระบบเพิ่ม | 44 |
| รูปที่ 6.1 แผนภูมิแสดงการเปลี่ยนแปลงค่าเฉลี่ยเวลาที่ใช้ในการค้นคืนต่อคำร้อง | 54 |



ความเป็นมาและปัญหา

งานเผยแพร่ข้อมูลข่าวสารในปัจจุบันสามารถกระทำได้หลากหลายรูปแบบขึ้นอยู่กับสื่อที่เลือกใช้ เช่น โทรทัศน์, วิทยุ, หนังสือพิมพ์, วารสาร ฯลฯ ระบบเครือข่ายเป็นอีกทางเลือกหนึ่ง เครือข่ายที่มีผู้นิยมเลือกใช้มากที่สุดได้แก่ เครือข่ายอินเทอร์เน็ต (Internet) โดยผ่านบริการเวิลด์ไวด์เว็บ (World Wide Web) ซึ่งแสดงผลโดยโปรแกรมค้นผ่านเว็บ (Web Browser) โดยข้อมูลที่ต้องการเผยแพร่จะถูกจัดเตรียมให้อยู่ในรูปของแฟ้มเว็บเพจ (Web Page) ปัญหาอย่างหนึ่งที่จะเกิดขึ้นตามมาคือ ถ้าแฟ้มเว็บเพจมีจำนวนมากขึ้น และผู้ใช้งานต้องการที่จะสืบค้นข้อมูลที่ต้องการอาจมีความยุ่งยาก หรือเสียเวลามากขึ้น ดังนั้นเพื่อแก้ปัญหานี้จึงได้มีการสร้างโปรแกรมค้นหา (Search Engine) ขึ้นเพื่อใช้สืบค้นข้อมูลที่ต้องการ

ปัจจุบันมีโปรแกรมค้นหาที่ใช้กันอยู่มากมาย อาทิเช่น โปรแกรมเนสเคปพับลิชชิงแอนด์เอเจนต์ (Netscape Publishing & Agent) ซึ่งเป็นส่วนหนึ่งของโปรแกรมเนสเคปเอ็นเตอร์ไพรส์เซิร์ฟเวอร์ (Netscape Enterprise Server) ของบริษัท เนสเคป, โปรแกรมอินเด็กซ์เซิร์ฟเวอร์ (Index Server) ซึ่งเป็นส่วนหนึ่งของโปรแกรมไอไอเอส (IIS ย่อมาจาก Internet Information System) ของบริษัท ไมโครซอฟต์, โปรแกรมอัลตาวิสตา (AltaVista) ของบริษัท ดิจิตอลอีควิปเมนต์ (Digital Equipment Corporation) โปรแกรมอัลตราซีคเซิร์ฟเวอร์ (Ultra Seek Server) ของบริษัท อินโฟซีค (Infoseek) ฯลฯ ซึ่งโดยส่วนใหญ่จะมีปัญหามากเมื่อใช้ร่วมกับตัวอักษรหรัสภาษาไทย เหตุผลหนึ่งคือ ภาษาไทยมีลักษณะการเขียนที่เป็นเอกลักษณ์เฉพาะตัว โดยแต่ละคำในประโยคจะเขียนติดกันไม่เว้นวรรคเหมือนภาษาอังกฤษ ซึ่งอาจก่อให้เกิดปัญหาในกรณีที่ต้องการจะค้นหาคำภาษาไทยที่อยู่ภายในประโยคโดยให้โปรแกรมค้นหาที่มีอยู่ในระบบเวิลด์ไวด์เว็บ ผลที่ได้รับคือ คำบางคำอาจหาพบ ในขณะที่บางคำอาจหาไม่พบ หรือผลลัพธ์อาจไม่ตรงตามที่ต้องการ เช่น

ต้องการค้นหาคำว่า 'ลิตร' สำหรับแฟ้มที่มีข้อความคือ

แฟ้มที่ 1 มีข้อความ "วันนี้ราคาน้ำมันลิตรละ 12 บาท" อยู่

แฟ้มที่ 2 มีข้อความ "ผมต้องการผลิตรายการโทรทัศน์" อยู่

จะเห็นได้ว่าทั้ง 2 แฟ้มมีคำว่า "ลิตร" ปรากฏอยู่ ซึ่งผลลัพธ์จากการค้นหาที่ตรงตามวัตถุประสงค์ที่ต้องการค้นหาควรจะมีเพียงแฟ้มที่ 1 เท่านั้น

จากปัญหาข้างต้น ถ้าเพียงจะอาศัยประสิทธิภาพของโปรแกรมสืบค้นเพียงอย่างเดียว อาจทำให้ไม่ได้ผลลัพธ์ตามที่ต้องการก็ได้

วัตถุประสงค์ของการวิจัย

เพื่อศึกษาและพัฒนาระบบที่จะสนับสนุนการค้นคืนบทความวิทยานิพนธ์ระดับมหาบัณฑิตของจุฬาลงกรณ์มหาวิทยาลัย โดยจะช่วยลดปัญหาการสืบค้นข้อมูลส่วนที่เป็นภาษาไทยให้น้อยลง

ขอบเขตของการวิจัย

1. ข้อมูลสำหรับนำเข้าระบบ คือ เพิ่มข้อความที่บันทึกรายงานบทความวิทยานิพนธ์ระดับมหาบัณฑิตของจุฬาลงกรณ์มหาวิทยาลัย ตั้งแต่ปีการศึกษา พ.ศ. 2538 ถึง ปี พ.ศ. 2540 โดยข้อมูลจะถูกบันทึกในรูปแบบอธิบายตนเอง (Self-describing)
2. ขั้นตอนวิธี (Algorithm) สำหรับการจัดทำดัชนีคำสำหรับประโยคภาษาไทยจะเป็นไปตามข้อกำหนดของวิทยานิพนธ์ ระบบการค้นคืนข้อความภาษาไทยโดยใช้เพิ่มข้อมูลผกผัน^[3]
3. โปรแกรมค้นหา SWISH-E^[6] ที่ผ่านการปรับปรุงเพื่อให้สามารถประมวลผลตัวอักษรรหัสภาษาไทย
4. โปรแกรมสนับสนุนสำหรับเครื่องบริการ (Server)
 - 4.1 ระบบปฏิบัติการวินโดวส์เอ็นทีตั้งแต่รุ่น 4.0 ขึ้นไป ที่สามารถประมวลผลตัวอักษรรหัสภาษาไทย
 - 4.2 ตัวบริการเว็บ (Web Server) ที่สนับสนุนมาตรฐาน โปรแกรมติดต่อประสานร่วมเกตเวย์ หรือ ซีจีไอ (CGI ย่อมาจาก Common Gateway Interface) ด้วยโปรแกรมภาษาเพิร์ล (Perl)
 - 4.3 ตัวแปลโปรแกรมภาษาซี^[12] และภาษาเพิร์ล^[10]
5. โปรแกรมสนับสนุนสำหรับเครื่องรับบริการ (Client)
 - 5.1 ระบบปฏิบัติการที่สามารถใช้โปรแกรมค้นผ่านเว็บในการแสดงผลภาษาไทยได้
 - 5.2 โปรแกรมค้นผ่านเว็บที่สามารถแสดงผลรหัสภาษาไทย แสดงกรอบ (Frame) และประมวลผลบทคำสั่งภาษาจาวา (Java Script)

ขั้นตอนการวิจัย

1. ศึกษาและคัดเลือกโปรแกรมค้นหาที่สามารถประมวลผลตัวอักษรรหัสภาษาไทยได้
2. ศึกษาขั้นตอนวิธีสำหรับการจัดทำดัชนีคำภาษาไทยจากวิทยานิพนธ์ ระบบการค้นคืนข้อความภาษาไทยโดยใช้เพิ่มข้อมูลผกผัน
3. ศึกษาโครงสร้างของเพิ่มข้อความรายงานบทความวิทยานิพนธ์ระดับมหาบัณฑิตของจุฬาลงกรณ์มหาวิทยาลัย
4. ออกแบบและพัฒนาระบบสนับสนุนการค้นคืนบทความวิทยานิพนธ์ผ่านเครือข่าย

5. ทดสอบระบบและประเมินประสิทธิภาพ
6. สรุปผลการวิจัยและข้อเสนอแนะ

ประโยชน์ที่คาดว่าจะได้รับ

1. โปรแกรมสนับสนุนการค้นคืนและแสดงผลสำหรับเพิ่มข้อความภาษาไทยโดยแปลงให้อยู่ในรูปของเพิ่มข้อมูลตามมาตรฐานภาษาเฮกซ์ที่เอ็มแอล (HTML ย่อมาจาก Hypertext Markup Language)
2. เป็นแนวทางในการพัฒนาโปรแกรมค้นหาสำหรับข้อมูลภาษาไทย



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

แนวคิดและทฤษฎีที่เกี่ยวข้อง

ในการออกแบบและพัฒนาระบบสนับสนุนการค้นคืนและแสดงผลข้อมูลจากแฟ้มข้อความจะต้องอาศัยความรู้และความเข้าใจในเรื่องต่างๆ ดังนี้คือ

1. ลักษณะโครงสร้างของข้อมูลที่อยู่ในแฟ้มข้อความที่ต้องการจัดเก็บ, ค้นคืนและแสดง
2. การจัดเตรียมข้อมูลเพื่อค้นหาและแสดงผล
3. การรับข้อคำถามเพื่อค้นหา
4. การแสดงผลข้อมูลที่ค้นได้

ดังนั้นในบทนี้จะเป็นการอธิบายสิ่งที่เกี่ยวข้องกับการออกแบบและพัฒนาระบบ ฯ ดังนี้

1. โครงสร้างแฟ้มข้อมูล อธิบายส่วนประกอบของข้อมูลที่ถูกจัดเก็บอยู่ในแฟ้มว่ามีรูปแบบการจัดเก็บอย่างไร
2. ขั้นตอนวิธีที่ใช้จัดทำดัชนีค่าที่จะถูกเรียกใช้เพื่อประมวลผลแฟ้มข้อมูลในขั้นตอนการจัดเตรียมข้อมูลเพื่อค้นหา และ ประมวลผลข้อความที่รับจากผู้ใช้เพื่อค้นหา
3. เวิลด์ไวด์เว็บและการสื่อสารสนเทศ อธิบายสภาพแวดล้อมของระบบ ฯ การเชื่อมต่อและสื่อสารระหว่างตัวบริการเว็บกับซีจีไอ และอธิบายขั้นตอนการทำงานโดยทั่วไปของโปรแกรมค้นหา

โครงสร้างแฟ้มข้อมูล ^[9]

แฟ้มข้อมูลประกอบด้วยระเบียบของข้อมูล เขตข้อมูล และส่วนข้อมูล ซึ่งความหมายของระเบียบ และเขตข้อมูลนี้ในทางตรรกะ เขตข้อมูล คือ หน่วยเล็กที่สุดที่สามารถจะแบ่งข้อมูลออกตามส่วนสำคัญ เช่น ชื่อ, นามสกุล, อายุ เป็นต้น ส่วนระเบียบ คือ กลุ่มของเขตข้อมูลที่เกี่ยวข้องกันมารวมกันเป็นข้อมูลหนึ่งหน่วย เช่น ระเบียบของพนักงานในองค์กรจะประกอบด้วยเขตข้อมูล ชื่อ , นามสกุล , อายุ , ที่อยู่ เป็นต้น

โดยในแฟ้มข้อมูลจะต้องประกอบด้วยระเบียบอย่างน้อย 1 ระเบียบ และ 1 ระเบียบจะต้องประกอบด้วยเขตข้อมูลอย่างน้อย 1 เขต

แนวปฏิบัติในการจัดการเกี่ยวกับเขตข้อมูล สามารถแบ่งวิธีการปฏิบัติหลักใหญ่ๆ ได้ 4 วิธีคือ

1. กำหนดความยาวให้แต่ละเขตข้อมูลไว้ล่วงหน้า
2. ระบุความยาวของเขตข้อมูลไว้ ณ ตำแหน่งแรกของเขตข้อมูล
3. ใส่ตัวอักษรพิเศษคั่น (Delimiter) ไว้ ณ จุดสิ้นสุดของแต่ละเขตข้อมูล
4. กำหนดนิพจน์ "ชื่อเขตข้อมูล = เนื้อหาข้อมูล" เพื่อระบุค่าของแต่ละเขตข้อมูล

วิธีที่ 1. กำหนดความยาวให้แต่ละเขตข้อมูลไว้ล่วงหน้า

วิธีนี้มีข้อดีคือ การจัดเก็บและเรียกคืนข้อมูลจะไม่ค่อยยุ่งยาก ข้อเสียคือ ถ้าเป็นข้อมูลประเภทที่มีความยาวไม่แน่นอน ถ้าต้องการให้สามารถบันทึกข้อมูลได้ครบถ้วนจะต้องกำหนดความยาวของเขตข้อมูลไว้ให้เพียงพอ ปัญหาที่เกิดขึ้นคือ ถ้าข้อมูลหนึ่งในเขตข้อมูลเดียวกันมีความยาวน้อยกว่าอีกข้อมูลหนึ่งจะทำให้เสียเนื้อที่เก็บข้อมูลแบบไม่มีประโยชน์ ดังนั้นวิธีนี้จะเหมาะสมสำหรับข้อมูลในเขตข้อมูลเดียวกันที่มีความยาวไม่แตกต่างกันมากนัก

วิธีที่ 2. ระบุความยาวของเขตข้อมูลไว้ ณ ตำแหน่งแรกของเขตข้อมูล

วิธีนี้เก็บค่าความยาวที่แท้จริงของข้อมูลในเขตข้อมูลไว้ที่ตำแหน่งแรกของเขตข้อมูล โดยค่าที่ใช้เก็บจะขึ้นอยู่กับขนาดของส่วนที่จะใช้เก็บความยาว เช่น ถ้ากำหนดขนาด 1 ไบต์ ความยาวสูงสุดของข้อมูลที่จะเก็บได้จะไม่เกิน 255 ตัวอักษร

วิธีที่ 3. ใส่ตัวอักษรพิเศษคั่นไว้ ณ จุดสิ้นสุดของแต่ละเขตข้อมูล

วิธีนี้จะใช้ตัวอักษรพิเศษหรือกลุ่มของตัวอักษรพิเศษ ซึ่งจะต้องไม่ปรากฏอยู่ในข้อมูลมาคั่น ณ จุดสิ้นสุดของแต่ละเขตข้อมูล

วิธีที่ 4. กำหนดนิพจน์ "ชื่อเขตข้อมูล = เนื้อหาข้อมูล" เพื่อระบุค่าของแต่ละเขตข้อมูล

วิธีนี้จะมีจุดเด่นกว่าวิธีอื่นๆ คือแต่ละเขตข้อมูลจะอยู่ในรูปที่สามารถอธิบายความหมายของข้อมูล หรือ อธิบายตนเอง (Self-describing) ซึ่งจะมีประโยชน์เกี่ยวกับการจัดการโครงสร้างและตรวจสอบความผิดพลาดของข้อมูลได้ง่าย เช่น ชื่อเขตข้อมูลที่ไม่สอดคล้องกับเนื้อหาข้อมูล อย่างไรก็ตามวิธีนี้ก็ยังมีข้อเสียคือ สลับเปลืองเนื้อที่ที่จะใช้เก็บข้อมูลส่วนที่เป็นชื่อเขตข้อมูล

จากวิธีที่กล่าวมา อาจจะนำมาใช้ร่วมกันก็ได้ เช่น ใช้วิธีที่ 3 ร่วมกับวิธีที่ 4 โดยระบุ

"ชื่อเขตข้อมูล = เนื้อหาข้อมูล" แล้วคั่นด้วยตัวอักษรพิเศษ

ตัวอย่าง

FIRSTNAME=สมศักดิ์|SURNAME=แช่ตั้ง|...

ใช้ FIRSTNAME และ SURNAME เป็นชื่อเขตข้อมูล และใช้ '|' เป็นตัวอักษรคั่น

ขั้นตอนวิธีที่ใช้จัดทำดัชนีคำ ^[3]

ขั้นตอนวิธีที่ใช้จะมาจากแนวคิดการจัดทำดัชนีคำโดยใช้พจนานุกรม เมื่อนำข้อความภาษาไทยผ่านขั้นตอนวิธีนี้ผลลัพธ์ที่ได้จะถูกแบ่งออกเป็น 2 กลุ่มคือ กลุ่มคำที่มีในพจนานุกรม และ กลุ่มของตัวอักษรที่ไม่มีในพจนานุกรม

กลุ่มของตัวอักษรที่ไม่มีในพจนานุกรมจะถูกส่งไปตัดคำโดยใช้กฎ ผลที่ได้คือ พยางค์ ดังนั้น ถ้านำคำที่มีอยู่พจนานุกรมที่ได้ครั้งแรก ร่วมกับพยางค์ที่ได้จากการตัดคำโดยใช้กฎ โดยคั่นแต่ละคำหรือพยางค์ด้วยช่องว่าง ก็สามารถนำผลลัพธ์ทั้งหมดมาจัดทำดัชนีเพื่อใช้ในการค้นหาได้ต่อไป

ตัวอย่างข้อความ “นายเจมส์มาร์ตินต้องการผลิตรายการโทรทัศน์”

เมื่อผ่านขั้นตอนวิธีจะได้ผลลัพธ์ดังนี้

คำที่มีในพจนานุกรม = { นาย,ต้องการ,การผลิต,รายการ,โทรทัศน์,ต้อง,ผลิต }

พยางค์ที่ไม่มีในพจนานุกรม = { เจมส์,มาร์,ติน }

ดังนั้นเมื่อเรียงเรียงใหม่จะได้ข้อความที่จะนำไปสร้างเพิ่มดัชนีคือ

“นาย เจมส์ มาร์ ติน ต้อง ต้องการ การผลิต ผลิต รายการ โทรทัศน์”

เว็ลด์ไวด์เว็บและการสื่อสารสารสนเทศ

เว็ลด์ไวด์เว็บเป็นระบบสื่อสารและสารสนเทศที่นิยมใช้กันมากบนอินเทอร์เน็ต ซึ่งมีการสื่อสารข้อมูลในลักษณะของการรับและให้บริการ (Client/Server) และมีโปรแกรมค้นผ่านเว็บเป็นตัวรับบริการเว็บ (Web Client) ซึ่งช่วยให้สามารถค้นหาเอกสารต่างๆ ที่เก็บอยู่บนคอมพิวเตอร์ในรูปแบบที่เรียกว่า ข้อความหลายมิติ (Hypertext) ได้สะดวกขึ้น โดยใช้หลักการอ้างอิงแหล่งที่อยู่

เว็ลด์ไวด์เว็บถูกพัฒนามาจากแนวความคิดของนายทิม เบอร์เนิร์ส ลี (Tim Berners Lee) และนักวิทยาศาสตร์จากห้องวิจัยของศูนย์วิจัยนิวเคลียร์ยุโรป เมืองเจนีวา ประเทศสวิตเซอร์แลนด์ โดยมีวัตถุประสงค์เพื่อรวบรวมข่าวสารสารสนเทศในอินเทอร์เน็ตให้เป็นกลุ่มก้อนและสามารถเชื่อมโยงถึงกันได้ทั่วโลก

ความสามารถพิเศษของเว็ลด์ไวด์เว็บคือ สามารถให้บริการทั้งข้อความ ภาพ เสียงหรือภาพเคลื่อนไหว ซึ่งรวมเรียกว่า สื่อหลายมิติ (Hypermedia) ที่สำคัญคือ ง่ายต่อการใช้งาน สามารถสร้างส่วนเชื่อมโยง (Link) จากเอกสารปัจจุบันไปยังเอกสารอื่นได้

ส่วนประกอบของเว็ลด์ไวด์เว็บประกอบด้วย

- เครื่องบริการและตัวบริการเว็บ ทำหน้าที่ให้บริการสารสนเทศตามที่ได้รับคำขอจากโปรแกรมค้นผ่านเว็บ
- เครื่องรับบริการและโปรแกรมค้นผ่านเว็บ ทำหน้าที่เป็นตัวประสานระหว่างผู้ใช้กับตัวบริการเว็บ

- เครื่องข่าย ทำหน้าที่รับ-ส่งข้อมูลระหว่างเครื่องบริการและเครื่องรับบริการตามมาตรฐานที่ตกลงกันล่วงหน้าซึ่งเรียกว่า โทกนทวิธิ (Protocol)

เนื่องจากเว็ลด์ไวด์เว็บเป็นระบบสื่อสารหนึ่งบนอินเตอรเนต การเรียกใช้จึงต้องอยู่ในรูปแบบมาตรฐานการเรียกใช้บริการของอินเตอรเนตที่เรียกว่า ยูอาร์แอล (URL ย่อมาจาก Uniform Resource Locator) ซึ่งมีรูปแบบ ดังนี้

ชื่อโทกนทวิธิ://ชื่อเครื่อง:หมายเลขช่องทาง/ชื่อสารบบ/ชื่อแฟ้ม

| | |
|-----------------------|---|
| ชื่อโทกนทวิธิ | เป็นการบอกว่าต้องการเข้าถึงข้อมูลโดยใช้โทกนทวิธิชนิดใด โดยปกติจะเป็น http |
| ชื่อเครื่อง (Host) | ระบุเครื่องที่เก็บข้อมูล โดยอาจเป็นชื่อโดเมน หรือ เลขที่อยู่ไอพี |
| หมายเลขช่องทาง (Port) | เป็นตัวเลขเฉพาะซึ่งระบุบริการที่ขอจากเครื่องให้บริการ (ถ้าไม่ระบุจะถูกกำหนดตามมาตรฐานของโทกนทวิธิ เช่น ช่องทาง 21 สำหรับ ftp ช่องทาง 70 สำหรับ gopher ช่องทาง 80 สำหรับ http เป็นต้น) |
| ชื่อสารบบ (Path) | ระบุตำแหน่งของข้อมูลที่เก็บอยู่ในเครื่องนั้นๆ |
| ชื่อแฟ้ม (File name) | ระบุชื่อแฟ้มข้อมูลที่ต้องการเข้าถึง (ถ้าไม่ระบุจะใช้ชื่อแฟ้มที่กำหนดค่าโดยปริยายจากตัวบริการเว็บ) |

ตัวอย่างของยูอาร์แอล <http://www.cp.eng.chula.ac.th/faculty/index.html>

http คือ โทกนทวิธิรูปแบบหนึ่ง

www.cp.eng.chula.ac.th คือ ชื่อโดเมนของภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

faculty คือ ตำแหน่งที่เก็บแฟ้มข้อมูล

index.html คือ ชื่อแฟ้ม ปกติถ้าไม่ระบุ โดยส่วนใหญ่ค่าโดยปริยายของชื่อแฟ้มคือ index.htm, index.html, default.htm, default.html

โปรแกรมค้นผ่านเว็บในเว็ลด์ไวด์เว็บสามารถเชื่อมต่อกับแหล่งข้อมูลได้หลายลักษณะหรือหลายโทกนทวิธิ เช่น

- HTTP (HyperText Transfer Protocol) สำหรับรับส่งข้อมูลแบบข้อความหลายมิติ

- FTP (File Transfer Protocol) สำหรับโอนย้ายแฟ้มข้อมูล
- GOPHER สำหรับค้นหาและแสดงสารสนเทศร่วมกัน โดยการใช้ระบบของเมนู
- FILE สำหรับเปิดแฟ้มข้อมูล
- NEWS (Network News Transfer Protocol , NNTP) สำหรับอ่านข่าวจากยูสเน็ต (Usenet หรือ User's Network)
- WAIS (Wide Area Information Server) สำหรับค้นหาข้อมูลด้วยเว็ส
- TELNET สำหรับเข้าใช้เครื่องระยะไกล

รายละเอียดบางส่วนของเครือข่ายอินเทอร์เน็ตและเวิลด์ไวด์เว็บที่เกี่ยวข้องกับงานวิจัย

1. เกณฑ์วิธีขนส่งข้อความหลายมิติ หรือ เฮชทีทีพี (HTTP ย่อมาจาก HyperText Transfer Protocol)

เฮชทีทีพี เป็นเกณฑ์วิธีที่ถูกออกแบบมาสำหรับการกระจายข้อมูลข่าวสารบนเวิลด์ไวด์เว็บ ถือเป็นเกณฑ์วิธีหลักที่ตัวบริการเว็บจะใช้ติดต่อกับโปรแกรมค้นผ่านเว็บ

เฮชทีทีพี เริ่มใช้งานครั้งแรกด้วย รุ่น 0.9 เมื่อปี ค.ศ.1990 โดยกำหนดให้ใช้กับการรับส่งข้อมูลแบบตัวอักษรเท่านั้น ต่อมามีการพัฒนาเป็นรุ่น 1.0 เมื่อปี ค.ศ. 1992 โดยมีการนำมาตรฐานส่วนขยายสื่อประสมในระบบอินเทอร์เน็ตแบบเอนกประสงค์หรือไมม์ (MIME ย่อมาจาก Multipurpose Internet Multimedia Extension) ซึ่งเป็นมาตรฐานสำหรับรับส่งข้อมูลแบบไบนารีร่วมกับรหัสแอสกี ทำให้สามารถรับส่งข้อมูลแบบสื่อประสมได้ จึงทำให้เวิลด์ไวด์เว็บได้รับความนิยมอย่างรวดเร็ว ปัจจุบันเฮชทีทีพีกำลังถูกพัฒนาให้เป็นรุ่น 1.1 อย่างไรก็ตามรูปแบบการติดต่อกันระหว่างตัวบริการเว็บและโปรแกรมค้นผ่านเว็บยังคงมาตรฐานรุ่น 1.0

เฮชทีทีพี มีการทำงานในรูปแบบร้องขอและตอบรับ โดยโปรแกรมค้นผ่านเว็บจะเป็นผู้เริ่มการติดต่อกับการส่งคำร้องขอไปยังตัวบริการเว็บ โดยระบุสารสนเทศที่จำเป็นไว้ที่ส่วนหัวของคำร้องขอ ส่วนภายในคำร้องขอจะบรรจุข้อมูลที่จะบอกให้ตัวบริการเว็บทราบถึงความต้องการ รวมถึงเงื่อนไขหรือข้อมูลที่จะส่งร่วมไปด้วยหรือไม่ก็ได้ ส่วนตัวบริการเว็บจะตอบกลับมาด้วยคำตอบรับ โดยระบุสารสนเทศที่จำเป็นไว้ที่ส่วนหัวของคำตอบรับ ในกรณีที่เป็นการร้องขอเอกสาร ส่วนที่ต่อจากส่วนหัวของคำตอบรับจะเป็นข้อมูลของเอกสารที่โปรแกรมค้นผ่านเว็บต้องการ หลังจากข้อมูลถูกส่งหมดแล้วการติดต่อกันนั้นๆ จะสิ้นสุดลง รายละเอียดของสารสนเทศของคำร้องขอและคำตอบรับสามารถดูจากตารางที่ 2.1

รูปแบบการส่งข้อมูลของส่วนหัวคำร้องขอ

```
Request_Method URL HTTP_Protocol_Version
{RequestHeaderField}
...
{เว้นหนึ่งบรรทัดเพื่อบอกให้ทราบว่าจบส่วนของ Header}
ข้อมูลที่ส่ง..
```

บรรทัดแรกจะเป็นบรรทัดคำร้อง ซึ่งจะถูกแบ่งเป็น 3 ส่วนได้แก่

- วิธีการส่งคำร้องขอ
- ยูอาร์แอล
- รุ่นของ เฮททีทีพี ที่ใช้งาน

วิธีการส่งคำร้องขอ

สำหรับ เฮททีทีพี รุ่น 1.0 และ 1.1 วิธีส่งคำร้องขอ ที่สามารถกระทำได้ คือ “Get”, “Head”, “Post”, “Put”, “Delete”, “Link” และ “Unlink” แต่โดยปกติจะใช้ 3 วิธี คือ “Get”, “Head” และ “Post”

- วิธีส่งแบบ “Get” เป็นวิธีปกติที่ใช้ติดต่อ โดยข้อมูลจะถูกบันทึกไว้ที่ตัวแปรสภาพแวดล้อม ชื่อ QUERY_STRING
- วิธีส่งแบบ “Head” ใช้ในกรณีที่ต้องการดูคุณสมบัติของเอกสาร โดยไม่ต้องมีการส่งเอกสารนั้นๆ มาด้วยทำให้ทำงานได้เร็วกว่าวิธีส่งแบบ “Get” มักใช้ในโปรแกรมค้นผ่านเว็บที่มีการใช้หน่วยความจำแคช หรือ ตัวบริการแทน (Proxy Server) ในกรณีที่ต้องการตรวจสอบว่าเอกสารนั้นๆ มีการแก้ไขเปลี่ยนแปลงหรือไม่ ถ้าไม่มีการเปลี่ยนแปลงก็จะนำข้อมูลในแคชมาแสดงโดยไม่จำเป็นต้องเรียกข้อมูลมาใหม่ หรือหากมีการเปลี่ยนแปลงก็จะใช้วิธีส่งแบบ “Get” ขอเอกสารนั้นมาใหม่
- วิธีส่งแบบ “Post” ข้อมูลที่ต้องการส่งจะอยู่ถัดจากส่วนหัว โดยบันทึกลงหน่วยรับมาตรฐาน โดยไม่มีการบอกจุดจบของข้อมูล แต่สามารถตรวจสอบได้จากตัวแปรสภาพแวดล้อม ชื่อ CONTENT_LENGTH ซึ่งจะบอกขนาดข้อมูลที่ส่งมาหน่วยเป็นไบต์

ถัดจากบรรทัดคำร้องขอจะเป็นเขตข้อมูลส่วนหัวของคำร้องขอ ซึ่งจะเป็นรายละเอียดต่างๆ ที่ต้องการส่งไปยังตัวบริการเว็บซึ่งอาจมีหลายบรรทัดขึ้นอยู่กับข้อมูลที่ต้องการส่ง

| ส่วนหัว | การทำงาน |
|----------------|---|
| Accept | ชนิดของข้อมูลที่โปรแกรมค้นผ่านเว็บที่สามารถรับและเข้าใจได้ เช่น Accept: text,audio,images |
| Accept-Charset | แจ้งให้ทราบว่าโปรแกรมค้นผ่านเว็บต้องการให้ส่งเอกสารที่ใช้ ชุดตัวอักษร (Character set) แบบใด เช่น Accept-Charset: iso-8859-1 |
| Allow | แจ้งให้โปรแกรมค้นผ่านเว็บทราบถึงวิธีการส่งคำร้องขอ ที่ยอมให้ใช้ได้ เช่น Allow: GET,HEAD |
| ContentLength | ขนาดของข้อมูลที่ส่งเป็นไบต์ เช่น ContentLength: 1234 |
| Content-Type | ชนิดของข้อมูลที่ส่ง เช่น Content-type: text/html |
| Date | วัน/เวลา แบบ GMT ที่ทำการส่งข้อมูล เช่น Date: Fri, 20 Nov 1998 12:34:45 GMT |
| Expire | วัน/เวลา แบบ GMT ที่ข้อมูลนี้หมดอายุ เช่น Expire: Thu, 31 Dec 1998 23:59:59 GMT |
| Last-Modified | วัน/เวลาที่เอกสารนั้นๆ ได้รับการแก้ไขครั้งล่าสุด เช่น Last-Modified: Tue, 1 Dec 1998 23:59:59 GMT |
| Location | ตัวบริการเว็บใช้บอกให้ทราบถึงตำแหน่งที่ถูกต้องของเอกสารที่ต้องการ กรณีใช้ร่วมกับรหัสสถานะเป็น 3XX จะหมายถึงการเปลี่ยนทิศทาง (Redirect) โดยอัตโนมัติ |
| MIME-Version | รุ่นของโมเมที่ใช้ เช่น MIME-Version: 1.0 |
| Server | ชนิดและรุ่นของตัวบริการเว็บ เช่น Server:Apache/1.3.3 |
| User-Agent | ชนิดและรุ่นของโปรแกรมค้นผ่านเว็บ เช่น User-Agent:Mozilla4.5[en](Win98;l) |

ตารางที่ 2.1 แสดงรายละเอียดของส่วนหัวเขตข้อมูลของคำร้องขอ/คำตอบรับ (บางส่วน)

หลังจากที่ตัวบริการเว็บได้รับข้อมูลส่วนหัวคำร้องขอแล้ว ตัวบริการเว็บจะทำการตรวจสอบสิ่งที่โปรแกรมค้นผ่านเว็บต้องการและสร้างส่วนหัวคำตอบรับเพื่อส่งกลับไปยังโปรแกรมค้นผ่านเว็บ

รูปแบบของส่วนหัวคำตอบรับ มีดังนี้

```

PROTOCOL/VERSIONStatus_CodeStatus_Description
{ResponseHeaderField}

```

...

{เว้นหนึ่งบรรทัดเพื่อบอกให้ทราบว่าจบส่วนของหัว}

เอกสารหรือไฟล์ที่ส่งไปยังโปรแกรมค้นผ่านเว็บ

บรรทัดแรกจะแบ่งออกเป็น 3 ส่วนได้แก่

- **PROTOCOL/VERSION** จะแทนที่ด้วยรุ่นของเฮชทีทีพีที่ใช้ในปัจจุบัน เช่น ระบุ "HTTP/1.0" หมายถึง เฮชทีทีพี รุ่น 1.0
- **Status_Code** แสดงรหัสสถานะเป็นค่าที่บอกสถานะการตอบสนองความต้องการของโปรแกรมค้นผ่านเว็บว่าสำเร็จหรือไม่ กลุ่มของค่ารหัสสถานะจะประกอบด้วยตัวเลข 3 หลัก
 - รหัสที่ขึ้นต้นด้วย 1 หมายถึง สารสนเทศ (เริ่มใช้กับ เฮชทีทีพี/1.1)
 - รหัสที่ขึ้นต้นด้วย 2 หมายถึง การร้องขอประสบความสำเร็จ
 - รหัสที่ขึ้นต้นด้วย 3 หมายถึง มีการเปลี่ยนทิศทาง
 - รหัสที่ขึ้นต้นด้วย 4 หมายถึง มีข้อผิดพลาดที่เกิดขึ้นจากโปรแกรมค้นผ่านเว็บ
 - รหัสที่ขึ้นต้นด้วย 5 หมายถึง มีข้อผิดพลาดจากตัวบริการเว็บ
- **Status_Description** สำหรับอธิบายรหัสสถานะ

ปัจจุบันเว็บไซต์เว็บได้รับความนิยมอย่างมากจึงได้มีการปรับปรุงประสิทธิภาพของ เฮชทีทีพี ให้ดีขึ้นเรื่อยๆ จากรุ่น 1.0 ที่มีปัญหาเรื่องของการรับส่งข้อมูลที่ไม่จำเป็น ทำให้ปริมาณข้อมูลที่วิ่งบนเครือข่ายเพิ่มสูงขึ้น มาเป็นรุ่น 1.1 ที่ได้รับการออกแบบเพื่อลดปัญหานี้โดยการเพิ่มประสิทธิภาพการใช้แชนแนลและตัวบริการแทนมากขึ้น ต่อมาได้มีความคิดที่จะพัฒนา เฮชทีทีพี ใหม่โดยใช้ชื่อว่า "HyperText Transfer Protocol Next Generation" หรือ HTTP-NG ซึ่งนอกเหนือขอบเขตของงานวิจัยนี้ ผู้ที่สนใจสามารถศึกษาได้จากเว็บไซต์เว็บคอนซอร์เทียม (World Wide Web Consortium) โดยค้นจากยูอาร์แอล <http://www.w3.org/Protocols/HTTP-NG/>

2. เฮชทีเอ็มแอล (HTML ย่อมาจาก HyperText Mark-up Language)

เฮชทีเอ็มแอล เป็นภาษาสำหรับสร้างข้อความหลายมิติบนเว็บไซต์เว็บ ภาษา เฮชทีเอ็มแอล นอกจากจะแสดงข้อมูลที่เป็นข้อความแล้วยังสามารถแสดงผลได้อีกหลายรูปแบบ เช่น ภาพนิ่ง, ภาพเคลื่อนไหว หรือ เสียง โดยอาศัยตามมาตรฐานไม่มี

เฮชทีเอ็มแอล เป็นส่วนหนึ่งของ ภาษาเอสจีเอ็มแอล (SGML ย่อมาจาก Standard Generalized Markup Language) ซึ่งกำหนดขึ้นโดย องค์การระหว่างประเทศว่าด้วยการมาตรฐาน หรือ ไอโซ (ISO ย่อมาจาก International Standards Organization) ภาษาเอสจีเอ็มแอลนี้ได้มีการพัฒนามาจากภาษาจีเอ็มแอล (GML ย่อมาจาก Generalized Markup Language) ของบริษัท ไอบีเอ็ม ทั้งจีเอ็มแอลและเอสจีเอ็มแอลถูกออกแบบโดย มีวัตถุประสงค์เพื่อสร้างระบบเอกสาร ตั้ง

แต่การเชื่อมโยงเอกสาร การแสดงผลทางหน่วยส่งออกมาตรฐาน เช่น จอภาพ เครื่องพิมพ์ โดยที่ไม่ขึ้นอยู่กับระบบปฏิบัติการหรือฮาร์ดแวร์

การควบคุมการแสดงผลของเอกสารของเฮททีเอ็มแอลจะกระทำผ่านรหัสเฮททีเอ็มแอล (HTML markup codes หรือ tags) ที่จะเป็นตัวกำหนดว่าจะให้โปรแกรมค้นผ่านเว็บแสดงเอกสารนั้นในรูปแบบใด การกำหนดรหัสเฮททีเอ็มแอลจะอยู่ระหว่างเครื่องหมาย "<" และ ">" เช่น

```
<Head>      กำหนดว่าข้อความที่ถัดไปจะถูกกำหนดเป็นส่วนหัวของเอกสารจนกว่าจะถึง
              </Head> ซึ่งหมายถึงสิ้นสุดส่วนหัวของเอกสาร
<P>         ใช้สำหรับกำหนดย่อหน้าของเอกสาร
```

ปัจจุบันเฮททีเอ็มแอลถูกกำหนดออกมาแล้ว 4 รุ่น ตั้งแต่ 0-3 โดยที่ทุกรุ่นที่ออกมาใหม่ จะครอบคลุมถึงรุ่นที่ออกมาแล้วด้วย ดังนั้นจึงไม่ต้องคำนึงถึงการเปลี่ยนแปลงการจัดรูปแบบใหม่เพื่อรองรับการเปลี่ยนแปลงรุ่นของเฮททีเอ็มแอลเลย

โดยปกติรูปแบบของเฮททีเอ็มแอล จะถูกแบ่งออกเป็น 3 ส่วนคือ

- ส่วนอารัมภบท (Prologue) เพื่อบอกระบุเอกลักษณ์ของเอกสารเฮททีเอ็มแอล เช่น HTML-DTD(Data Type Definition) ใช้บอกรุ่นของเฮททีเอ็มแอล ที่ใช้ในเอกสารเฮททีเอ็มแอลให้กับโปรแกรมค้นผ่านเว็บทราบ ตัวอย่าง

```
<!doctype HTML public "-//W30 //DTD W3 HTML 3.0//EN">
```

- ส่วนหัวเอกสาร ใช้กำหนดส่วนเพิ่มที่จะครอบคลุมทั้งเอกสาร เช่น
 - รหัส Title สำหรับแจ้งให้โปรแกรมค้นผ่านเว็บแสดงข้อความขอบบนของวินโดว์ เช่น <title>ระบบสนับสนุนการค้นคืนข้อมูลวิทยานิพนธ์ผ่านเครือข่าย</title>
 - รหัส Base ใช้ระบุจุดเริ่มต้นของการเชื่อมโยงที่ใช้ภายในเอกสาร มีประโยชน์ในกรณีที่มีการเชื่อมโยงเอกสาร โดยที่ระบุเพียงชื่อเอกสารที่ต้องการเชื่อมโยงเท่านั้น แล้วมีการนำเอกสารนี้ไปไว้ที่อื่น ซึ่งถ้าไม่ได้ระบุรหัส Base ไว้ล่วงหน้า โปรแกรมค้นผ่านเว็บก็จะหาเอกสารที่ต้องการเชื่อมโยงไม่พบ ตัวอย่าง

เอกสารของ Own Host ระบุ

```
<Head>
<Base HREF=http://www.ownhost.com/>
</Head>
<body>

</body>
```

รหัส Base จะแจ้งให้โปรแกรมค้นผ่านเว็บทราบว่าที่อยู่หลักของเอกสารคือ www.ownhost.com เมื่อมีการคัดลอกเพิ่มนี้ไปที่ www.OtherHost.com เวลาแสดงผลรูปภาพ [bluedot.gif](http://www.ownhost.com/images/bluedot.gif) โปรแกรมค้นผ่านเว็บจะไปนำเพิ่มมาจาก <http://www.ownhost.com/images/bluedot.gif> มาแสดง

- รหัส Meta ใช้ระบุสารสนเทศลงในแฟ้มเอกสาร สารสนเทศบางอย่างจะถูกนำไปใช้สำหรับระบุชนิดของเอกสาร การจัดทำดัชนี การจัดหมวดหมู่ ส่วนประกอบของรหัส Meta ประกอบด้วย
 - Http-equiv สำหรับกำหนดสารสนเทศส่วนต่อรับเซชที่ที่พี ตัวอย่าง


```
<meta http-equiv="Content-type" content="text/html">
```

 ระบุชี้เฉพาะว่าเป็นเอกสารเซชที่เอ็มแอล แบบข้อความ
 - Name กำหนดชื่อของสารสนเทศ ถ้าไม่ระบุจะถูกกำหนดให้เป็น Http-equiv ตัวอย่าง


```
<meta name="Author" content="Sopon Tongkasem">
```
 - Content แสดงรายละเอียดสารสนเทศ
- ส่วนข้อมูลของเอกสาร สำหรับแสดงผลรายละเอียดของเอกสารตามรูปแบบของรหัสเซชที่เอ็มแอล ผ่านโปรแกรมค้นผ่านเว็บ

3. ซีจีไอ (CGI ย่อมาจาก Common Gateway Interface)

คือ เกณฑ์วิธีหนึ่งบนเซชที่ที่พี ที่ทำหน้าที่ควบคุมการสื่อสารระหว่างตัวบริการเว็บและโปรแกรมค้นผ่านเว็บแบบ 2 ทิศทาง (Two way communications) ซีจีไอ ถูกสร้างขึ้นมาเพื่อเพิ่มความสามารถให้กับเซชที่ที่พี เนื่องจากเซชที่ที่พีไม่เปิดโอกาสให้กับตัวรับบริการเว็บและตัวบริการเว็บโต้ตอบกันในลักษณะ 2 ทิศทาง

ซีจีไอ ประกอบด้วย เว็บเพจที่มีการสร้างฟอร์มเซชที่เอ็มแอลสำหรับรับข้อมูลจากผู้ใช้ และโปรแกรมที่ถูกเขียนขึ้นมาตามมาตรฐานของซีจีไอ เรียกว่า บทคำสั่งซีจีไอ (CGI Script) โดยที่ทั้งเว็บเพจและบทคำสั่งซีจีไอจะถูกเก็บไว้ที่เครื่องบริการ ควบคุมโดยตัวบริการเว็บ

ภาษาที่นิยมใช้เขียนโปรแกรมบทคำสั่งซีจีไอ ได้แก่ ภาษาเพิร์ล ภาษาซี ภาษาปาสคาล ภาษาวิซวลเบสิก ฯลฯ

วิธีการส่งข้อมูลจากโปรแกรมค้นผ่านเว็บไปยังโปรแกรมบทคำสั่งซีจีไอ

เมื่อผู้ใช้งานปุ่มควบคุมการตอบรับ (Submit) โปรแกรมค้นผ่านเว็บจะทำการจัดรูปแบบข้อมูลแต่ละเขตข้อมูลโดยระบุ ชื่อเขตข้อมูลร่วมกับข้อมูลที่ผ่านมาการเข้ารหัสแล้ว รวมเป็นคู่ โดยคั่นระหว่างคู่

ด้วยเครื่องหมาย '&' หลังจากนั้นก็จะจัดรวมกับชื่อพารามิเตอร์ชื่ออื่น เพื่อส่งให้กับตัวบริการเว็บในรูปแบบยูอาร์แอล ดังนี้

ชื่อเกณฑ์วิธี://ชื่อเครื่อง:หมายเลขช่องทาง/ชื่อสารบบ/ชื่อพารามิเตอร์ชื่ออื่น?name1=value1&name2=value2&...

หลังจากนั้นตัวบริการเว็บจะส่งข้อมูลให้กับโปรแกรมพารามิเตอร์ชื่ออื่น ตามที่ระบุใน ยูอาร์แอล วิธีการส่งข้อมูลของตัวบริการเว็บ สามารถทำได้ 2 แบบคือ วิธีส่งแบบ "Get" และวิธีส่งแบบ "Post" ตามมาตรฐานของเฮชทีทีพีที่กล่าวมาแล้ว

เมื่อโปรแกรมพารามิเตอร์ชื่ออื่นได้รับข้อมูลจากตัวบริการเว็บแล้ว จะทำการแยกเขตข้อมูลและข้อมูลออกเพื่อถอดรหัสข้อมูล หลังจากนั้นค่อยนำข้อมูลที่ได้ออกไปประมวลผลต่อไป

วิธีการส่งข้อมูลจากโปรแกรมพารามิเตอร์ชื่ออื่นไปยังโปรแกรมคั่นผ่านเว็บ

เมื่อโปรแกรมพารามิเตอร์ชื่ออื่นประมวลผลเรียบร้อยแล้วจะส่งผลลัพธ์มายังผู้ใช้โดยผ่านตัวบริการเว็บทางหน่วยส่งออกมาตรฐาน โดยจะต้องมีการแจ้งให้ตัวบริการเว็บทราบว่าข้อมูลที่ส่งจะเป็นประเภทใด ตัวอย่าง

1. Content-type: text/html
- 2.
3. <HTML>
4. ... ส่วนข้อมูลที่ต้องการส่ง
5. </HTML>

ดังตัวอย่าง เป็นการส่งข้อมูลแบบเฮชทีเอ็มแอล โดยระบุ Content-type เป็นแบบ text/html แล้วเว้น 1 บรรทัด หลังจากนั้นค่อยตามด้วยข้อมูลที่ต้องการส่ง (ตั้งแต่บรรทัดที่ 3 เป็นต้นไป)

4. ชุดตัวอักษร (Character sets)

ตัวอักษรที่ใช้สำหรับแสดงผลบนเฮชทีเอ็มแอล จะใช้ตัวอักษรแบบ 8 บิต (รหัสแอสกี) ดังนั้นจึงสามารถบันทึกด้วยตัวอักษรรหัสภาษาไทยได้ นอกจากนี้เฮชทีเอ็มแอลยังได้เพิ่มเติมรูปแบบการเขียนตัวอักษรอีก 2 ประเภทคือ

1. อ้างอิงเอนทิตีด้วยตัวอักษร (Character Entity References)

รูปแบบนี้จะใช้ชุดตัวอักษรแทนตัวอักษร 1 ตัว ชุดตัวอักษรนี้จะเริ่มต้นด้วยเครื่องหมาย & และสิ้นสุดด้วยเครื่องหมาย ; เช่น ! แทนตัวอักษรรหัสแอสกีที่ 161 , ¢ แทนตัวอักษรรหัสแอสกีที่ 162 เป็นต้น

2. อ้างอิงเอนทิตีด้วยตัวเลข (Numerical Entity References)

รูปแบบนี้จะระบุตัวอักษรด้วยชุดตัวเลขตามรหัสแอสกี ชุดตัวเลขนี้จะนำหน้าด้วยเครื่องหมาย &# และสิ้นสุดด้วยเครื่องหมาย ; เช่น ¡ ¢ เป็นต้น

ถ้าตัวอักษรในเอกสารเฮกซ์ที่เอ็มแอลถูกบันทึกด้วยรูปแบบ อ้างอิงเอนทิตีด้วยตัวอักษร หรือ อ้างอิงเอนทิตีด้วยตัวเลข จะทำให้ไม่สามารถใช้ขั้นตอนวิธีจัดทำดัชนีคำภาษาไทยได้โดยตรง แม้ว่าตัวอักษรที่แสดงผลผ่านโปรแกรมค้นผ่านเว็บจะแสดงผลเป็นภาษาไทยก็ตาม

5. โปรแกรมค้นหา

เป็นเครื่องมือช่วยค้นหาข้อความหลายมิติที่อยู่บนเวปไซด์ไวด์เว็บ ขั้นตอนการทำงานของโปรแกรมค้นหามีดังนี้

1. จัดเตรียมแฟ้มดัชนี (Indexing) ได้แก่ การอ่านข้อมูลเว็บเพจ กรองคำที่ไม่ใช้ทำดัชนีออก เช่น คำที่เป็นรหัสเฮกซ์ที่เอ็มแอล หลังจากนั้นก็ตรวจสอบคำว่าสามารถทำดัชนีได้หรือไม่ ตัดคำบางคำที่คาดว่าจะมีมากเกินไป โดยไม่นำไปทำดัชนีคำ เช่น a, an, the เป็นต้น จัดเก็บรายละเอียดอื่นที่เกี่ยวข้องกับคำที่ใช้ทำดัชนี เช่น ชื่อแฟ้มหรือยูอาร์แอลของเว็บเพจ , title , meta data, ตำแหน่งของคำที่อยู่บนเว็บเพจ ซึ่งในการจัดเก็บรายละเอียดนี้จะขึ้นอยู่กับโปรแกรมค้นหาด้วยว่าถูกออกแบบมาให้เก็บข้อมูลส่วนใดบ้าง
2. ค้นหา ซึ่งมักจะมาจากโปรแกรมบทคำสั่งซีจีไอที่เชื่อมโยงกับเว็บเพจที่จัดทำขึ้นเพื่อให้ผู้ใช้ป้อนข้อความ (Query Term) ที่ผู้ใช้ต้องการให้ค้นหา หลังจากที่โปรแกรมค้นหาได้ทำการประมวลผลข้อความเรียบร้อยแล้ว ถ้าค้นพบ โปรแกรมค้นหาจะส่งคืนข้อมูลที่ค้นได้โดยเรียงลำดับตามความสำคัญของคำหรือข้อความที่อยู่บนเว็บเพจ

โปรแกรมค้นหานับเป็นเครื่องมือสำคัญอย่างยิ่งในระบบการสืบค้นข้อมูลจากอินเทอร์เน็ต เทคโนโลยีของการค้นหายังเป็นหัวข้อที่มีผู้สนใจพัฒนาอย่างมากโดยเฉพาะการประยุกต์ใช้หลักทางปัญญาประดิษฐ์เข้าไปช่วยในการทำดัชนีและค้นหา โปรแกรมค้นหาที่สามารถสืบค้นคำในภาษาไทยก็นับว่าเป็นสิ่งที่ต้องการอย่างมาก ซึ่งก็ยังเป็นสิ่งที่ต้องพัฒนาเพื่อให้ได้โปรแกรมค้นหาที่สามารถสืบค้นคำภาษาไทยได้อย่างสมบูรณ์ต่อไป

บทที่ 3

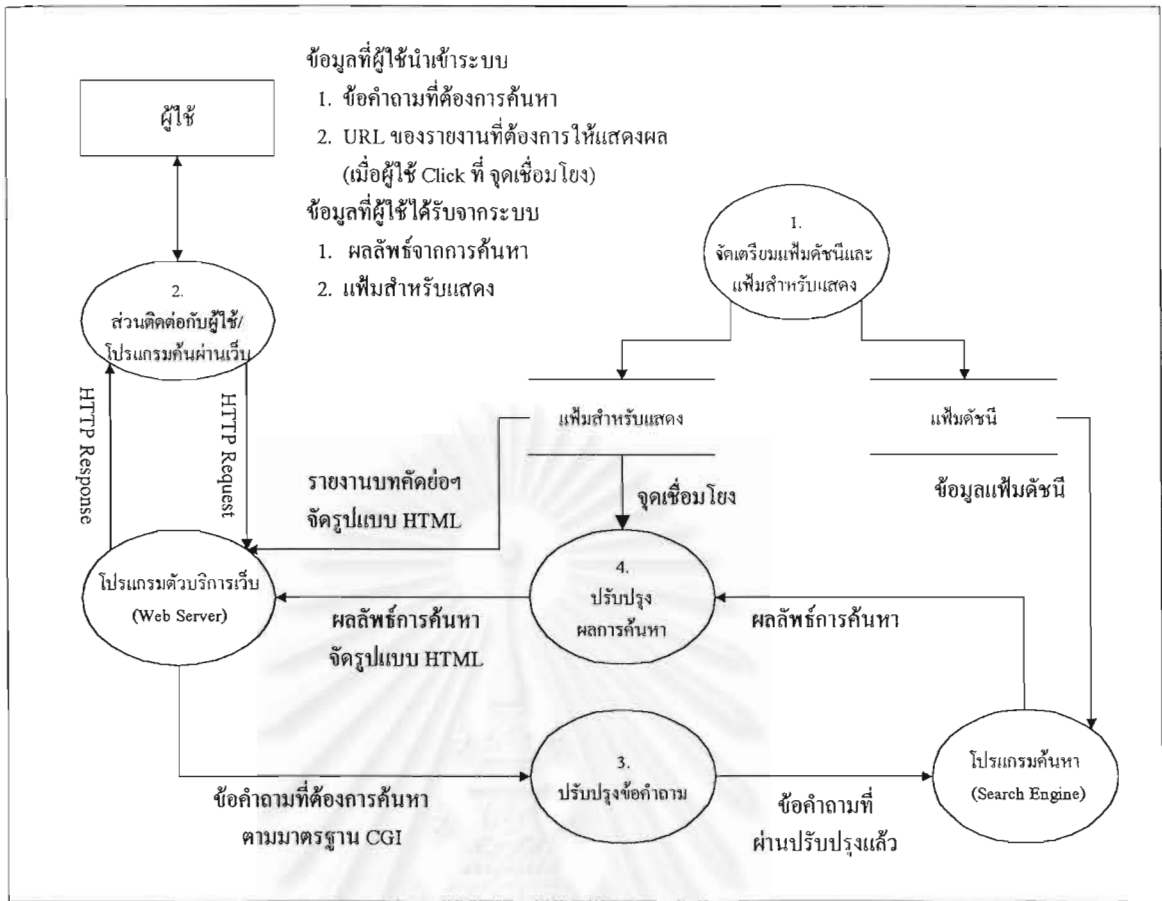
การวิเคราะห์และออกแบบระบบ

การสืบค้นข้อมูลข่าวสารไทยโดยผ่านบริการเว็ลด์ไวด์เว็บ โดยทั่วไปมักจะพบปัญหาที่เกิดขึ้นคือ โปรแกรมค้นหาที่จะใช้สำหรับสืบค้นไม่ได้ออกแบบให้รองรับกับข้อมูลภาษาไทย เนื่องจากภาษาไทยเป็นภาษาที่มีเอกลักษณ์เป็นของตนเอง การเรียบเรียงคำในแต่ละประโยคจะต่อเนื่องกันไปไม่มีการเว้นวรรคภายในประโยค ซึ่งต่างกับลักษณะภาษาของประเทศทางตะวันตกที่ส่วนใหญ่จะเป็นผู้พัฒนาโปรแกรมค้นหา โดยลักษณะภาษาของประเทศทางตะวันตกจะเป็นลักษณะคั่นแต่ละคำด้วยช่องว่าง การจัดทำแฟ้มดัชนีและการค้นคืนของโปรแกรมค้นหา จึงสามารถจัดทำดัชนีคำได้จากการตรวจสอบช่องว่างระหว่างคำ ซึ่งวิธีนี้จะใช้ไม่ได้ผลกับประโยคในภาษาไทย ด้วยเหตุนี้ ผู้วิจัยจึงขอเสนอแนวทางหนึ่งสำหรับการพัฒนาโปรแกรมสนับสนุนการค้นคืนและแสดงผลสำหรับเพิ่มข้อความภาษาไทย โดยพัฒนาขึ้นจากขั้นตอนวิธีจัดทำดัชนีคำจากงานวิจัยเรื่อง “ระบบการค้นคืนข้อความภาษาไทยโดยใช้เพิ่มข้อมูลผกผัน”^[3] โดยลักษณะของเพิ่มข้อความภาษาไทยที่จะนำเข้าสู่ระบบจะต้องอยู่ในรูปโครงสร้างแฟ้มแบบอธิบายตนเอง โดยใน 1 แฟ้มข้อความประกอบด้วยข้อมูลหลายระเบียบ แต่ละระเบียบจะประกอบด้วยชื่อเขตข้อมูลและค่าของเขตข้อมูลตั้งแต่ 1 รายการขึ้นไป เมื่อจบแต่ละระเบียบจะต้องมีตัวคั่นเพื่อบอกจุดสิ้นสุด ดังรูปแบบต่อไปนี้

| |
|---|
| (ระเบียบที่ 1) ชื่อเขตข้อมูล1: เนื้อหาข้อมูล1 ชื่อเขตข้อมูล2: เนื้อหาข้อมูล2 ชื่อเขตข้อมูล3: เนื้อหาข้อมูล3 .. <ตัวคั่นสิ้นสุดระเบียบ> |
| (ระเบียบที่ 2) ชื่อเขตข้อมูล1: เนื้อหาข้อมูล1 ชื่อเขตข้อมูล2: เนื้อหาข้อมูล2 ชื่อเขตข้อมูล3: เนื้อหาข้อมูล3 .. <ตัวคั่นสิ้นสุดระเบียบ> |
| ... |
| (ระเบียบที่ n) ชื่อเขตข้อมูล1: เนื้อหาข้อมูล1 ชื่อเขตข้อมูล2: เนื้อหาข้อมูล2 ชื่อเขตข้อมูล3: เนื้อหาข้อมูล3 .. <ตัวคั่นสิ้นสุดระเบียบ> |

แนวคิดในการออกแบบการทำงานของโปรแกรมสนับสนุนการค้นคืนและแสดงผลสำหรับเพิ่มข้อความสามารถแบ่งออกเป็น 4 ขั้นตอนหลักดังนี้ (ดูรูปที่ 3.1)

1. การจัดเตรียมแฟ้มดัชนีและแฟ้มสำหรับแสดง
2. การรับข้อความคำถามจากผู้ใช้เพื่อค้นหา
3. การปรับปรุงข้อความก่อนที่จะส่งไปโปรแกรมค้นหา
4. การปรับปรุงผลการค้นหาที่ได้รับจากโปรแกรมค้นหา



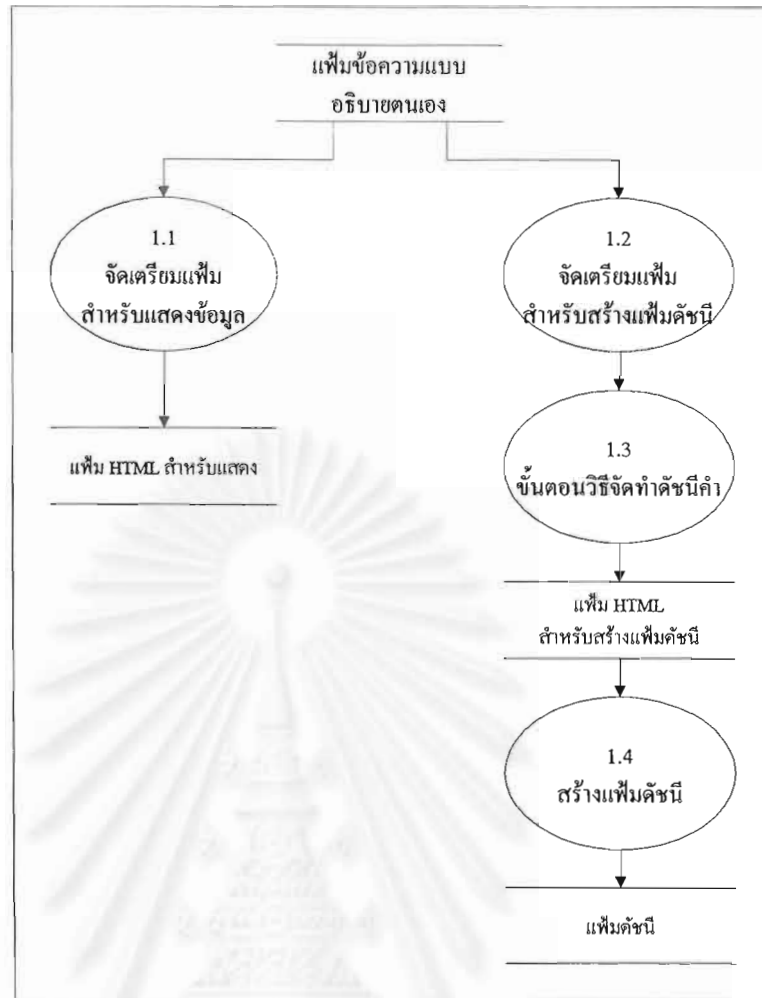
รูปที่ 3.1 แสดงการทำงานโดยรวมของระบบ

1. การจัดทำเตรียมแฟ้มดัชนีและแฟ้มสำหรับแสดง

แฟ้มดัชนี คือ แฟ้มที่โปรแกรมค้นหาต้องเรียกใช้เวลาที่ต้องการค้นหาคำ โดยปกติในแฟ้มดัชนีจะประกอบด้วย คำที่ถูกจัดทำดัชนีที่ถูกจัดเรียงตามขั้นตอนวิธีจัดเรียงของโปรแกรมค้นหาแต่ละชนิด และรายละเอียดบางอย่างที่สามารถแสดงให้ผู้ใช้ทราบว่าคำนั้นค้นมาจากเอกสารใดและในเอกสารนั้นเกี่ยวข้องกับเรื่องอะไรบ้าง แฟ้มดัชนีโดยปกติจะถูกสร้างโดยโปรแกรมค้นหา

แฟ้มสำหรับแสดง คือ แฟ้มที่ถูกสร้างมาจากข้อมูลที่อยู่ในแฟ้มข้อความที่มีโครงสร้างแบบอธิบายตนเอง

โดยปกติโปรแกรมค้นหาจะทำการสร้างแฟ้มดัชนีจากแฟ้มสำหรับแสดง แต่วิธีนี้จะใช้ไม่ได้ผลสำหรับเอกสารภาษาไทยเนื่องจากสาเหตุที่กล่าวมาแล้ว ดังนั้นจึงต้องมีการจัดทำเตรียมแฟ้มอีกชุดหนึ่ง โดยที่แฟ้มชุดนี้จะต้องผ่านขั้นตอนวิธีจัดทำดัชนีคำสำหรับข้อความภาษาไทยก่อน หลังจากนั้นจึงค่อยส่งไปโปรแกรมค้นหาเพื่อสร้างแฟ้มดัชนีต่อไป



รูปที่ 3.2 แสดงการจัดเตรียมเพิ่มดัชนีและเพิ่มสำหรับแสดง

จากรูปที่ 3.2 สามารถอธิบายได้ดังนี้

1.1 จัดเตรียมเพิ่มสำหรับแสดงข้อมูล

ประมวลผลเพิ่มข้อความ ฯ โดย แยกข้อมูลแต่ละระเบียนเขียนลงในแฟ้มใหม่ให้อยู่ในรูปแบบตามมาตรฐานภาษาเฮกซ์ที่เอ็มแอล จัดรูปแบบเพื่อให้ผู้ใช้อ่านได้ง่าย

1.2 จัดเตรียมเพิ่มสำหรับสร้างเพิ่มดัชนี

ประมวลผลเพิ่มข้อความ ฯ โดย แยกข้อมูลแต่ละระเบียนออกและจัดรูปแบบของข้อมูลในระเบียนใหม่ให้มีเนื้อหากระชับขึ้น ตัดส่วนที่ไม่จำเป็นต้องจัดทำดัชนีออก แทรกรายละเอียดบางอย่างที่จำเป็นต้องใช้ในการสร้างเพิ่มดัชนีลงไปแทน เช่น รหัส title , รหัส Meta tag

1.3 ขั้นตอนวิธีจัดทำดัชนีคำ

นำข้อมูลที่ได้จากข้อ 1.2 มาผ่านขั้นตอนวิธีจัดทำดัชนีคำ เขียนผลลัพธ์ที่ได้ลงแฟ้มใหม่

1.4 สร้างแฟ้มดัชนี

นำแฟ้มที่ได้จากข้อ 1.3 มาสร้างแฟ้มดัชนี โดยปกติจะถูกสร้างโดยโปรแกรมค้นหา หลังจากที่ได้จัดทำแฟ้มดัชนีเรียบร้อยแล้ว ก็สามารถลบแฟ้มที่ได้จากข้อ 1.3 นี้ได้

2. การรับข้อความจากผู้ใช้เพื่อค้นหา

เป็นการจัดทำส่วนติดต่อกับผู้ใช้ในรูปแบบของเว็บเพจที่ถูกแสดงด้วยโปรแกรมค้นหาผ่านเว็บ สำหรับให้ผู้ใช้ระบุข้อความที่ต้องการค้นหา ซึ่งจะอยู่ในรูปแบบฟอร์มของเฮกซ์ที่เต็มแอล ข้อความที่ผู้ใช้ป้อนเรียบร้อยแล้วจะถูกส่งไปยังโปรแกรมบทคำสั่งซีไอที่ตัวบริการเว็บ

ข้อความที่ได้รับจากผู้ใช้ประกอบด้วย ข้อความและตัวกระทำซึ่งระบุหรือไม่ก็ได้ ตัวกระทำที่สามารถใช้ได้จะขึ้นอยู่กับข้อกำหนดของโปรแกรมค้นหา

โดยปกติสำหรับโปรแกรมค้นหาทั่วไปการประมวลผลสำหรับตัวกระทำแต่ละชนิดก็จะแตกต่างกันไป

ตัวอย่าง

ข้อความ1 + ข้อความ2 หมายถึง กำหนดให้ค้นหาเอกสารที่มีทั้งข้อความ1 และ ข้อความ2

ข้อความ1 or ข้อความ2 หมายถึง กำหนดให้ค้นหาเอกสารที่มีทั้งข้อความ1 หรือ ข้อความ2

ตัวอย่างข้างต้น อ้างอิงตามข้อกำหนดของโปรแกรมค้นหา SWISH-E ที่ใช้ในงานวิจัยนี้ ดังนั้น สำหรับโปรแกรมค้นหาอื่นอาจแตกต่างไปจากข้างต้นก็ได้

3. การปรับปรุงข้อความที่รับจากผู้ใช้ก่อนที่จะส่งไปโปรแกรมค้นหา

ข้อความที่ตัวบริการเว็บได้รับจากขั้นตอนที่ 2 จะอยู่ในรูปของการเข้ารหัส ดังนั้นจึงต้องมีการถอดรหัสให้เป็นรหัสแอสกีก่อน จากนั้นนำไปแบ่งคำในกรณีที่มีข้อความภาษาไทย (โดยใช้ขั้นตอนวิธีเดียวกันกับขั้นตอนวิธีจัดทำดัชนีคำ)

สิ่งสำคัญอย่างหนึ่งในการพิจารณาเลือกโปรแกรมค้นหาที่จะนำมาใช้ นอกจากความสามารถประมวลผลตัวอักษรภาษาไทยได้แล้ว ยังต้องสามารถประมวลข้อความที่มีมากกว่า 1 ข้อความ โดยสามารถระบุตัวกระทำที่จะกำหนดให้ค้นหาเอกสารที่มีข้อความทั้งหมดใน 1 ข้อความนั้น เช่น ข้อความ "ข้อความ1 + ข้อความ2" โปรแกรมค้นหาจะต้องสามารถค้นหาเอกสารที่มีทั้งข้อความ1 และข้อความ2 ได้ (ตัวกระทำอาจเป็นอย่างอื่นนอกจาก + ก็ได้ เช่น ตัวกระทำ and แต่ลักษณะการประมวลผลจะต้องเป็นแบบเดียวกันคือค้นหาเอกสารที่มีทั้งข้อความ1 และข้อความ2) เหตุผลจากข้อพิจารณาข้างต้น มาจากความแตกต่างระหว่างคำในประโยคระหว่างภาษาไทยกับภาษาอังกฤษ

ตัวอย่าง เอกสารที่มีประโยค “สมเกียรติจัดรายการโทรทัศน์”

เมื่อผ่านขั้นตอนวิธีจัดทำดัชนีคำจะได้ “สมเกียรติ จัด รายการ โทรทัศน์”

ถ้าต้องการค้นหาข้อความที่มีข้อความ “รายการโทรทัศน์” เมื่อผ่านขั้นตอนวิธีแบ่งคำจะได้ข้อความใหม่ คือ “รายการ โทรทัศน์” โดยผลลัพธ์ที่ต้องการได้รับจากโปรแกรมค้นหาจะต้องได้รายชื่อแฟ้มเอกสารที่มีทั้งข้อความ “รายการ” และ “โทรทัศน์” ดังนั้นจะต้องมีการปรับปรุงข้อความข้างต้นใหม่เพื่อให้สอดคล้องกับตัวกระทำที่โปรแกรมค้นหาอนุญาตให้ใช้ได้ เช่น โปรแกรมค้นหากำหนดให้ใช้ตัวกระทำ + สำหรับการค้นหาเอกสารที่มีข้อความในข้อความมากกว่า 1 ข้อความ

จากข้อความ “รายการโทรทัศน์” เมื่อผ่านการปรับปรุงก็จะได้ “รายการ+โทรทัศน์” สำหรับค้นหาเอกสารที่มีทั้งข้อความ “รายการ” และ “โทรทัศน์”

4. การปรับปรุงผลการค้นหาที่ได้รับจากโปรแกรมค้นหา

ผลลัพธ์การสืบค้นของโปรแกรมค้นหาจะถูกส่งผ่านทางหน่วยส่งออกมาตรฐาน นำผลลัพธ์นี้มาจัดรูปแบบตามมาตรฐานเฮกซ์ที่เอ็มแอล เพิ่มจุดเชื่อมโยง (Anchor) เพื่อให้ผู้ใช้สามารถเรียกดูรายละเอียดจากแฟ้มสำหรับแสดงได้โดยตรงหรือไม่พบให้แสดงรายการเพื่อแจ้งผู้ใช้ทราบว่าไม่พบรายการที่ต้องการค้นหา

บทที่ 4

การพัฒนาระบบสำหรับจัดเตรียมเพิ่มข้อมูล

จากการวิเคราะห์และออกแบบระบบ ฯ ในบทที่ 3 ส่วนที่เกี่ยวกับการจัดเตรียมเพิ่มข้อมูลจะมีสิ่งที่ต้องพัฒนาตามลำดับดังนี้

1. พัฒนาโปรแกรมจัดทำดัชนีคำ
2. พัฒนาโปรแกรมจัดเตรียมเพิ่มข้อมูล

การพัฒนาโปรแกรมจัดทำดัชนีคำ

วัตถุประสงค์เพื่อให้สามารถแบ่งคำไทยสำหรับจัดทำดัชนีคำ โดยแบ่งการพัฒนาเป็น 2 ส่วนคือ

1. ฟังก์ชันสำหรับจัดทำดัชนีคำ สำหรับเรียกใช้โดยโปรแกรมจัดเตรียมเพิ่มข้อมูล
2. โปรแกรมแบ่งคำ สำหรับเรียกใช้โดยโปรแกรมจัดเตรียมข้อมูลคำถามก่อนที่จะส่งไปประมวลผลที่โปรแกรมค้นหา

การพัฒนาฟังก์ชันจัดทำดัชนีคำ

ฟังก์ชันจัดทำดัชนีคำนี้ถูกพัฒนาด้วยภาษาซี กำหนดให้เรียกใช้ฟังก์ชันที่พัฒนาจากงานวิจัย “ระบบการค้นคืนข้อความภาษาไทยโดยใช้เพิ่มข้อมูลผกผัน”^[3] ที่มีรูปแบบการเรียกใช้ดังนี้

```
void sepword( char *Target , char *Source );
```

Target คือ ที่พักสายอักขระ(String Buffer) สำหรับบันทึกผลลัพธ์

Source คือ สายอักขระ (String) ที่เก็บข้อความที่ต้องการจัดทำดัชนีคำ

รูปแบบการเรียกใช้ฟังก์ชันจัดทำดัชนีคำ

```
void thaiword( char *Target , char *Source );
```

Target คือ ที่พักสายอักขระที่ผ่านการจัดสรรหน่วยความจำแล้วสำหรับบันทึกผล

ลัพธ์

Source คือ สายอักขระที่เก็บข้อความที่ต้องการจัดทำดัชนีคำ

ขั้นตอนการทำงานของฟังก์ชันจัดทำดัชนีคำ

1. อ่านข้อมูลในเพิ่มพจนานุกรมบันทึกในหน่วยความจำ

2. ส่งค่าสายอักขระที่อยู่ในตัวแปร Source ไปประมวลผลที่ฟังก์ชัน sepword ที่ละชุดโดยแต่ละชุดจะถูกแยกโดยการตรวจสอบว่าไม่ใช่กลุ่มรหัสแอสกี ตัวอักษรภาษาไทยและน่าจะเป็นตัวอักษรที่เริ่มคำใหม่ (เงื่อนไขการตรวจสอบคัดมาจางานวิจัย ฯ เช่นกัน) ผลลัพธ์จากฟังก์ชัน sepword จะถูกบันทึกต่อท้ายกับค่าเดิม (ถ้ามี) กระบวนการนี้จะสิ้นสุดเมื่อจบค่าสายอักขระที่อยู่ในตัวแปร Source

รายละเอียดฟังก์ชัน

```
void thaiword(char *cTarget , char *cSource ) {
    char str[4096],ch;
    int i,n;
    if( !limit_Ready ) {
        initialize( "NewDict.txt" );
    }
    cTarget[0] = '\0';
    n = i=0;
    while( cSource[n] ) {
        ch=cSource[n++];

        if((((ch>='ก')&&(ch<='จ')) ||((ch>='เ')&&(ch<='อ'))&&(ch!='ฯ')&&(ch!='๗')) {
            str[i++]=ch;
        } else {
            if(i>0) {
                str[i]='\0';
                sepword( cTarget , str);
            }
            sprintf( cTarget , "%s%c",cTarget,ch );
            i=0;
        }
    }
    if(i>1) {
        str[i]='\0';
        sepword( cTarget , str);
    }
}
```

ตัวอย่าง การเรียกใช้ฟังก์ชัน

```
int main() {
    char source[100];
    char result[150];
    strcpy( source , "การจัดทำดัชนีคำ " );

    // เว้นช่องว่างหลังตัวอักษรสุดท้ายอย่างน้อย 2 ตัวอักษรเพื่อป้องกันการขดเซยสระ
    thaiword( result , source);
    printf("result = [%s]\n",result); // ผลลัพธ์ result = [การ จัดทำ ดัชนี คำ ]
}
```

การพัฒนาโปรแกรมแบ่งคำ

โปรแกรมแบ่งคำถูกพัฒนาด้วยภาษาซี กำหนดให้รับข้อมูลจากบรรทัดคำสั่ง (command line) การประมวลผลกำหนดให้ประมวลผลโดยเรียกใช้ฟังก์ชันจัดทำดัชนีคำ และเมื่อประมวลผลเรียบร้อยแล้ว จะส่งผลลัพธ์ออกหน่วยส่งออกมาตรฐาน

รูปแบบการเรียกใช้งาน

```
Tsepword.exe -w ข้อความ1 ข้อความ2 ... [-d เพิ่มพจนานุกรม]
```

-w สำหรับระบุข้อความที่ต้องการแบ่งคำ สามารถระบุต่อเนื่องกันไปได้ขึ้นอยู่กับขนาดของ Buffer ที่ใช้รับข้อมูล โดยที่ถ้าเป็นข้อความภาษาไทยให้ตามด้วยช่องว่างขนาด 2 ตัวอักษรเพื่อป้องกันการขีดเซยสระ

-d สำหรับระบุชื่อสารบบและชื่อเพิ่มพจนานุกรม ถ้าไม่ระบุโปรแกรมจะกำหนดค่าโดยปริยายคือ "newdict.txt" (กรณีเรียกใช้ผ่านบริการซีจีไอ จะต้องมีการระบุด้วย)

เนื่องจากการพัฒนาระบบถูกกำหนดให้ต้องสอดคล้องกับโปรแกรมค้นหา SWISH-E ดังนั้นผลลัพธ์จึงได้จัดเตรียมตามข้อกำหนดของ SWISH-E ที่อนุญาตให้ระบุตัวกระทำลงในข้อความ

รายละเอียดตัวกระทำของ SWISH-E

- ตัวกระทำ And ได้แก่

$s1$ and $s2$, $s1+s2$, $s1 s2$, ($s1 s2$) ค้นหาเอกสารที่มีข้อความทั้ง $s1$ และ $s2$

- ตัวกระทำ Or ได้แก่

$s1$ or $s2$ ค้นหาเอกสารที่มีข้อความ $s1$ หรือ $s2$

- ตัวกระทำ Not ได้แก่

Not $s1$ ค้นหาเอกสารที่ไม่มีข้อความ $s1$

ขั้นตอนการทำงานของโปรแกรมแบ่งคำ

1. ตรวจสอบอาร์กิวเมนต์ (argument) แยกส่วนข้อความและชื่อเพิ่มพจนานุกรม
2. ถ้ามีการระบุชื่อเพิ่มพจนานุกรม ให้เรียกฟังก์ชันสำหรับบันทึกข้อมูลพจนานุกรมลงหน่วยความจำ
3. ประมวลผลข้อความ โดยแยกข้อความออกเป็นชุดโดยตรวจสอบจากช่องว่าง ส่งข้อความแต่ละชุดไปประมวลผลที่ฟังก์ชันจัดทำดัชนีคำ (thaiword()) ตรวจสอบผลลัพธ์ว่ามีกร

เปลี่ยนแปลงจากข้อความที่ส่งหรือไม่ ถ้ามีการเปลี่ยนแปลงให้เพิ่มวงเล็บไว้หน้าและหลัง
ผลลัพธ์ ทั้งนี้เพื่อกำหนดขอบเขตเฉพาะให้โปรแกรมต้องค้นเอกสารที่มีคำทั้งหมดในผล
ลัพธ์เท่านั้น หลังจากนั้นบันทึกผลลัพธ์ เมื่อประมวลผลข้อความครบทุกชุดให้บันทึก
เครื่องหมาย “ไว้ที่หน้าและหลังของผลลัพธ์รวมทั้งหมดเพื่อระบุขอบเขต ส่งผลลัพธ์ทั้ง
หมดแสดงทางหน่วยส่งออกมาตรฐาน

ก่อนประมวลผล

กำหนดให้ข้อความ = stuv..

กำหนดให้ a เท่ากับผลลัพธ์ทั้งหมด

การประมวลผล

ประมวลผลข้อความทีละชุด = thaiword(r,s);

ถ้าผลลัพธ์ r = s1 s2 s3 .. sn

ใส่วงเล็บหน้าและหลัง จะได้ r = (s1 s2 s3 .. sn)

ถ้าผลลัพธ์ r = s

ไม่ต้องทำอะไรเพิ่ม

a = a + “ + r

ทำให้ครบทั้ง stuv..

เมื่อประมวลผลเรียบร้อย

ส่ง “a” ออกหน่วยส่งออกมาตรฐาน

ตัวอย่าง

#1 ข้อความ ระบบคอมพิวเตอร์และการสื่อสาร

ประมวลผล tsepword -w ระบบคอมพิวเตอร์และการสื่อสาร

ผลลัพธ์ = “(ระบบ คอมพิวเตอร์ และ การ สื่อสาร)”

#2 ข้อความ ระบบคอมพิวเตอร์+การสื่อสาร

ประมวลผล tsepword -w ระบบคอมพิวเตอร์+การสื่อสาร

ผลลัพธ์ = “(ระบบ คอมพิวเตอร์ + การ สื่อสาร)”

#3 ข้อความ ระบบคอมพิวเตอร์ + การสื่อสาร

ประมวลผล tsepword -w ระบบคอมพิวเตอร์ + การสื่อสาร

ผลลัพธ์ = “(ระบบ คอมพิวเตอร์) + (การ สื่อสาร)”

#4 ข้อความ ระบบคอมพิวเตอร์ and การสื่อสาร

ประมวลผล tsepword -w ระบบคอมพิวเตอร์ and การสื่อสาร

ผลลัพธ์ = “(ระบบ คอมพิวเตอร์) and (การ สื่อสาร)”

#5 .ข้อความ ระบบคอมพิวเตอร์ or การสื่อสาร

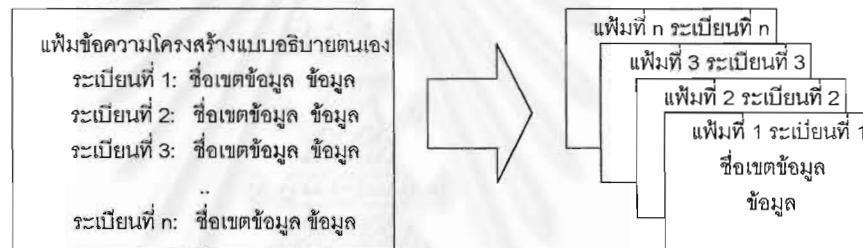
ประมวลผล tseword -w ระบบคอมพิวเตอร์ or การสื่อสาร

ผลลัพธ์ = "(ระบบ คอมพิวเตอร์) or (การ สื่อสาร)"

ผลลัพธ์ที่ได้จากข้อ #1 , #2 , #3 , #4 เมื่อผ่านโปรแกรมค้นหา SWISH-E จะได้ผลเช่นเดียวกัน คือ รายชื่อเพิ่มเอกสารที่มีคำว่า "ระบบ" , "คอมพิวเตอร์" , "การ" , "สื่อสาร" อยู่ในแฟ้มเดียวกัน ส่วนผลลัพธ์ที่ได้จากข้อ #5 จะได้รายชื่อเพิ่มเอกสารที่มีคำว่า "ระบบ" และ "คอมพิวเตอร์" หรือเพิ่มเอกสารที่มีคำว่า "การ" และ "สื่อสาร"

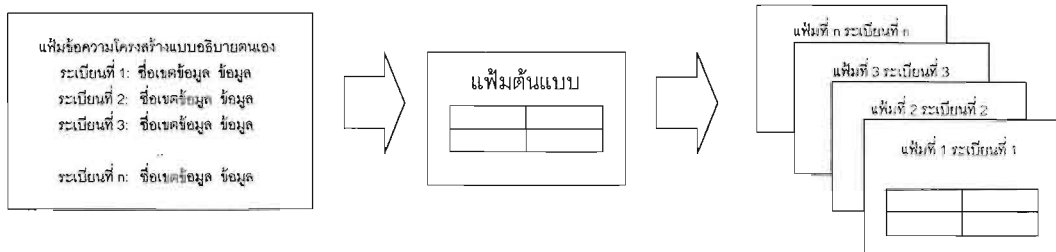
การพัฒนาโปรแกรมจัดเตรียมแฟ้มข้อมูล

จากแนวทางการวิเคราะห์และออกแบบระบบสำหรับจัดเตรียมแฟ้มข้อมูลในบทที่ 3 สามารถแสดงได้ดังรูปที่ 4.1



รูปที่ 4.1 แสดงรูปแบบการจัดเตรียมแฟ้มข้อมูล

เพิ่มข้อความโครงสร้างแบบอธิบายตนเอง จะเป็นแฟ้มรวมข้อมูลทั้งหมดทุกระดับ เมื่อผ่านการจัดการจะได้ผลลัพธ์เป็นแฟ้มย่อย ๆ แต่ละแฟ้มประกอบด้วยข้อมูล 1 ระดับที่ถูกจัดข้อมูลตามมาตรฐานภาษาเอชทีเอ็มแอล สาเหตุที่ต้องจัดแบ่งเป็นแฟ้มย่อยเนื่องจากผลลัพธ์ที่ได้จากโปรแกรมค้นหาจะเป็นชื่อแฟ้ม ดังนั้นถ้านำแฟ้มรวมข้อมูลทั้งหมดมานำเสนอเพียงแฟ้มเดียวการนำเสนอสิ่งที่ผู้ใช้งานต้องการก็จะกระทำลำบาก ในทางตรงกันข้ามถ้าแบ่งออกเป็นแฟ้มย่อยเวลานำเสนอก็เพียงแต่นำแฟ้มที่เกี่ยวข้องที่เป็นผลลัพธ์จากโปรแกรมค้นหามาแสดงเท่านั้น และเพื่อให้แฟ้มข้อมูลที่ได้จากการจัดเตรียมนี้มีรูปแบบไปในแนวทางเดียวกันและสะดวกในการจัดการ จึงได้เพิ่มส่วนการจัดรูปแบบข้อมูลผ่านแฟ้มต้นแบบ (Template file) ดังรูปที่ 4.2

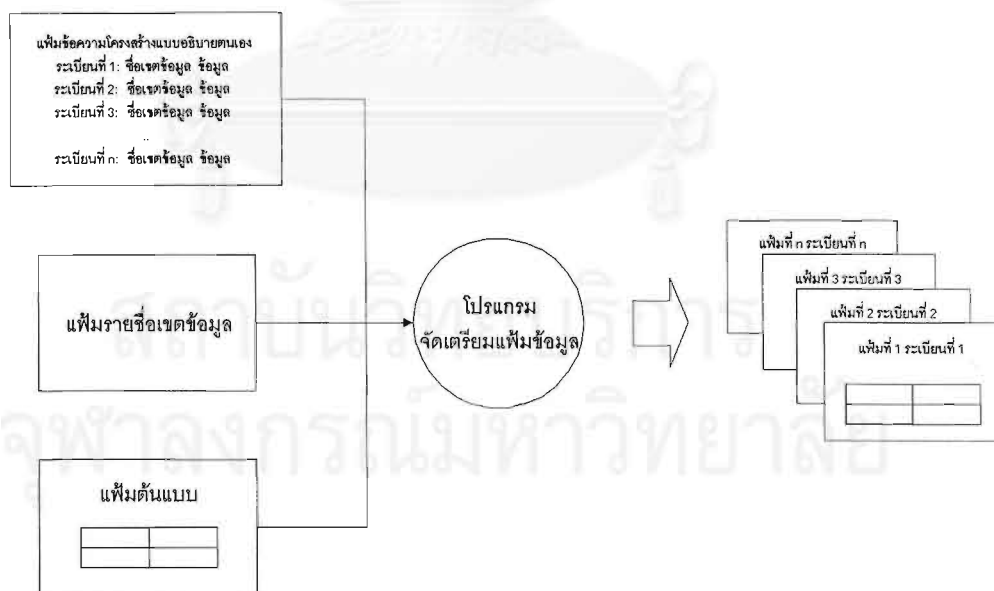


รูปที่ 4.2 แสดงรูปแบบการจัดเตรียมเพิ่มข้อมูลผ่านเพิ่มต้นแบบ

เพิ่มต้นแบบจะประกอบด้วยรหัสเขตที่เอ็มแอลและตำแหน่งของชื่อเขตข้อมูลที่ต้องการให้แทนค่าด้วยข้อมูลนั้นๆ ดังนั้นการควบคุมการแสดงข้อมูลจึงสามารถกระทำผ่านเพิ่มต้นแบบ (โดยสามารถสร้างจาก โปรแกรมบรรณาธิการทั่วไปสำหรับเพิ่มเขตที่เอ็มแอล)

โปรแกรมจัดเตรียมเพิ่มข้อมูลนี้ถูกพัฒนาด้วยภาษาซี (ดูรูปที่ 4.3) ประกอบด้วยสิ่งที่เกี่ยวข้องดังนี้

1. เพิ่มข้อความที่บันทึกข้อมูลที่มีโครงสร้างแบบอธิบายตนเอง
2. เพิ่มข้อความที่บันทึกรายชื่อเขตข้อมูลที่ใช้ในเพิ่มข้อ 1.
3. เพิ่มต้นแบบ



รูปที่ 4.3 แสดงสิ่งที่เกี่ยวข้องกับโปรแกรมจัดเตรียมเพิ่มข้อมูล

มีรูปแบบการเรียกใช้งาน

```
Gnmain.exe -t ชื่อแฟ้มหรือสถานที่เก็บ [-f ชื่อแฟ้ม] -p ชื่อแฟ้ม [-l ชื่อแฟ้ม]
```

- t เพิ่มข้อความที่บันทึกข้อมูลที่มีโครงสร้างแบบอธิบายตนเอง
- f เพิ่มรายชื่อเขตข้อมูล
- p เพิ่มต้นแบบ
- l เพิ่มสำหรับบันทึกรายละเอียดการทำงานแต่ละขั้นตอนของโปรแกรม (log file)

สิ่งที่ต้องจัดเตรียมก่อนการประมวลผลโปรแกรมจัดเตรียมเพิ่มข้อมูล คือ

1. เพิ่มรายชื่อเขตข้อมูล
2. เพิ่มต้นแบบ

การจัดเตรียมเพิ่มรายชื่อเขตข้อมูล

ใช้ระบุรายละเอียดที่เกี่ยวกับเขตข้อมูลที่ใช้ในเพิ่มข้อมูลที่มีโครงสร้างแบบอธิบายตนเอง รายละเอียดที่บันทึกในเพิ่มรายชื่อเขตข้อมูลจะแบ่งออกเป็น 3 ส่วน คือ

1. ส่วนควบคุม จะนำหน้าด้วย . (จุด) สามารถกำหนดได้ดังนี้ มีรูปแบบโดยทั่วไปคือ

```
.ตัวควบคุม ค่าของตัวควบคุม
```

รายละเอียดตัวควบคุมที่สามารถเรียกใช้ได้

| ตัวควบคุม | รายละเอียด |
|------------------|--|
| Field_delimiter | ระบุตัวอักษรที่ใช้คั่นระหว่างเขตข้อมูล (ความยาวสูงสุดไม่เกิน 10 ตัวอักษร) |
| Record_delimiter | ระบุตัวคั่นเมื่อจบแต่ละระเบียน (ความยาวสูงสุดไม่เกิน 25 ตัวอักษร) |
| Counter | กำหนดตัวนับที่จะเพิ่มขึ้นทุกครั้งที่มีการเรียกใช้ในเพิ่มต้นแบบ ตัวนับนี้จะถูกใช้เพื่อเป็นค่าเฉพาะ (Unique Value) เช่น ชื่อแฟ้มที่เป็นผลลัพธ์ของโปรแกรมจัดเตรียมเพิ่มข้อมูล รูปแบบของ Counter .Counter[:ความยาว] ค่าเริ่มต้น [,ค่าถัดไป 1 [,ค่าถัดไป 2, [,ค่าถัดไป n]]] - ถ้าไม่ระบุความยาว โปรแกรมจัดเตรียมเพิ่มข้อมูลจะกำหนดค่าโดยปริยาย = 4 - เมื่อมีการเรียกใช้ค่าตัวนับ โปรแกรมจะทำการอ่านค่าที่อยู่ทางขวาสุด และนำไปแทนค่า ณ ตำแหน่งที่กำหนดจากเพิ่มต้นแบบ และเพิ่มค่าขึ้นอีก 1 - เมื่อโปรแกรม ๙ ประมวลผลเรียบร้อยแล้ว โปรแกรม ๙ จะบันทึกค่าสุดท้ายที่มีการเรียกใช้ต่อท้ายบรรทัด .counter ไปเรื่อยๆ |

ตัวอย่าง

.field_delimiter ::

กำหนดให้ :: เป็นตัวคั่นระหว่างเขตข้อมูล เช่น

NAME::Sopon SURNAME::Tongkasem

.record_delimiter ~~

กำหนดให้ ~~ เป็นตัวคั่นเมื่อจบแต่ละระเบียน เช่น

ระเบียน 1 ~~ ระเบียน 2 ~~ ระเบียน 3 ~~

.counter:6 1

กำหนดให้ Counter เริ่มที่ 000001 มีความยาวเท่ากับ 6 เมื่อสิ้นสุดการประมวลผลมีการเรียกใช้ 5 ครั้ง โปรแกรมจะทำการบันทึกดังนี้

.counter:6 1,00006

2. ส่วนรายชื่อเขตข้อมูล

รูปแบบการบันทึก

ชื่อเขตข้อมูล [:ความยาว | { เริ่ม , จำนวน}] [,ส่วนขยาย]

- ชื่อเขตข้อมูล ความยาวสูงสุดไม่เกิน 99 ตัวอักษร โดยมีเงื่อนไข ห้ามขึ้นต้นด้วย '.' และต้องไม่มีตัวอักษร ':' หรือ ',' อยู่ในชื่อเขตข้อมูล
- ความยาว ถ้าไม่ระบุโปรแกรมจะบันทึกตามค่าเขตข้อมูล
ถ้าค่าความยาวเป็นบวก จะบันทึกโดยนับตั้งแต่ตัวอักษรแรก
ถ้าค่าความยาวเป็นลบ จะบันทึกตั้งแต่ตำแหน่งตัวอักษรสุดท้าย-ความยาว
- เริ่ม กำหนดตำแหน่งของแรกของค่าเขตข้อมูล (เริ่มตั้งแต่ 0)
- จำนวน กำหนดจำนวนตัวอักษรโดยนับจาก 'เริ่ม'
- ส่วนขยาย ใช้กำหนดรายละเอียดการจัดการเกี่ยวกับค่าของเขตข้อมูล
N หรือ n ลบตัวอักษรขึ้นบรรทัดใหม่ (new line) ออกจากค่าของเขตข้อมูล
T หรือ t ให้ผ่านค่าของเขตข้อมูลไปฟังก์ชันจัดทำดัชนีค่า

ตัวอย่าง ดูจากตัวอย่างในหัวข้อการจัดเตรียมเพิ่มเติมแบบ

ชื่อเขตข้อมูล ในกรณีที่เป็นภาษาอังกฤษจะไม่คำนึงว่าจะเป็นตัวอักษรพิมพ์เล็กหรือพิมพ์ใหญ่ โดยถ้ามีการระบุชื่อเขตข้อมูลซ้ำกัน โปรแกรมจะถือตามชื่อเขตข้อมูลแรกสุด

จำนวนเขตข้อมูลสูงสุดไม่เกิน 48 รายการ

3. ส่วนหมายเหตุ

ข้อความที่อยู่หลังเครื่องหมาย // โปรแกรม ฯ จะถือว่าเป็นหมายเหตุ

การจัดเตรียมเพิ่มเติมแบบ

รายละเอียดที่สามารถบันทึกในเพิ่มเติมแบบ จะแบ่งออกเป็น 4 ส่วนคือ

1. ส่วนที่ใช้สำหรับจัดรูปแบบและควบคุมการแสดงรายงานบนโปรแกรมค้นผ่านเว็บ
2. ส่วนชื่อเขตข้อมูล ใช้กำหนดตำแหน่งที่ต้องการให้แทนด้วยค่าของเขตข้อมูลในระเบียน ปัจจุบันขณะประมวลผล มีรูปแบบการเรียกใช้ดังนี้

```
%ชื่อเขตข้อมูล [:ความยาว | { เริ่ม , จำนวน} ] [ส่วนขยาย]%
```

ชื่อเขตข้อมูลจะอยู่ระหว่างเครื่องหมาย % .. % ถ้าไม่ระบุความยาวและส่วนขยาย โปรแกรมจะกำหนดค่าโดยปริยายตามรายการที่บันทึกในเพิ่มรายชื่อเขตข้อมูล ส่วนขยายจะกำหนดเช่นเดียวกับส่วนขยายที่บันทึกในเพิ่มรายชื่อเขตข้อมูล

ตัวอย่าง

ชื่อเขตข้อมูล IDNO มีค่าของเขตข้อมูล 3972306621

- | | | | |
|----|---------------------|----------------|--------------------|
| #1 | (เพิ่มเติมแบบ) | <P>เลขประจำตัว | %IDNO% |
| | (เพิ่มที่สร้างใหม่) | <P>เลขประจำตัว | 3972306621 |
| #2 | (เพิ่มเติมแบบ) | <P>เลขประจำตัว | %IDNO:4% |
| | (เพิ่มที่สร้างใหม่) | <P>เลขประจำตัว | 3972 |
| #3 | (เพิ่มเติมแบบ) | <P>เลขประจำตัว | %IDNO:(3,5)% |
| | (เพิ่มที่สร้างใหม่) | <P>เลขประจำตัว | 23066 |

3. ส่วนกำหนดชื่อเพิ่ม สำหรับกำหนดชื่อเพิ่มที่จะสร้างขึ้นใหม่ โดยจะต้องเป็นชื่อที่ไม่ซ้ำกับชื่อเพิ่มที่มีอยู่ กรณีที่ชื่อเพิ่มเกิดซ้ำกับชื่อเพิ่มที่มีอยู่ในระบบ โปรแกรมฯ จะทำการสร้างทับเพิ่มเดิม ชื่อเพิ่มนี้สามารถใช้ร่วมกับค่าของเขตข้อมูลและค่าของ Counter ที่ระบุในเพิ่มรายชื่อเขตข้อมูล การระบุส่วนกำหนดชื่อเพิ่ม ให้ระบุในส่วนหัวของรหัสเฮกซ์ที่เอ็มแอล (ระหว่าง <head> ... </head>) โดยส่วนกำหนดชื่อเพิ่มนี้จะไม่ถูกบันทึกในเพิ่มที่สร้างขึ้นใหม่ มีรูปแบบการใช้งานดังนี้

```
<meta name="TSCMD.writeto" content="ชื่อเพิ่ม">
```


ตัวอย่าง

- ชื่อเขตข้อมูล IDNO มีค่าของเขตข้อมูล 3972306621
- Counter กำหนดความยาว = 4 ค่าต่อไป = 0001
- ส่วนกำหนดที่ระบุชื่อเพิ่ม ในแฟ้มต้นแบบ

<meta name="TSCMD.writeto" content="%IDNO:4%%COUNTER%.htm">

- ผลลัพธ์จะได้ชื่อเพิ่ม 39720001.htm

4. ส่วนรายละเอียดประกอบอื่นๆ

ใช้บันทึกที่ร่วมกับค่าของเขตข้อมูลที่ต้องการบันทึกในแฟ้มที่จะสร้างขึ้นใหม่

ตัวอย่าง

เลขประจำตัว %IDNO% คณะ ภาควิชา

ขั้นตอนการทำงานของโปรแกรมจัดเตรียมแฟ้มข้อมูล (ดูรูปที่ 4.4 ประกอบ)

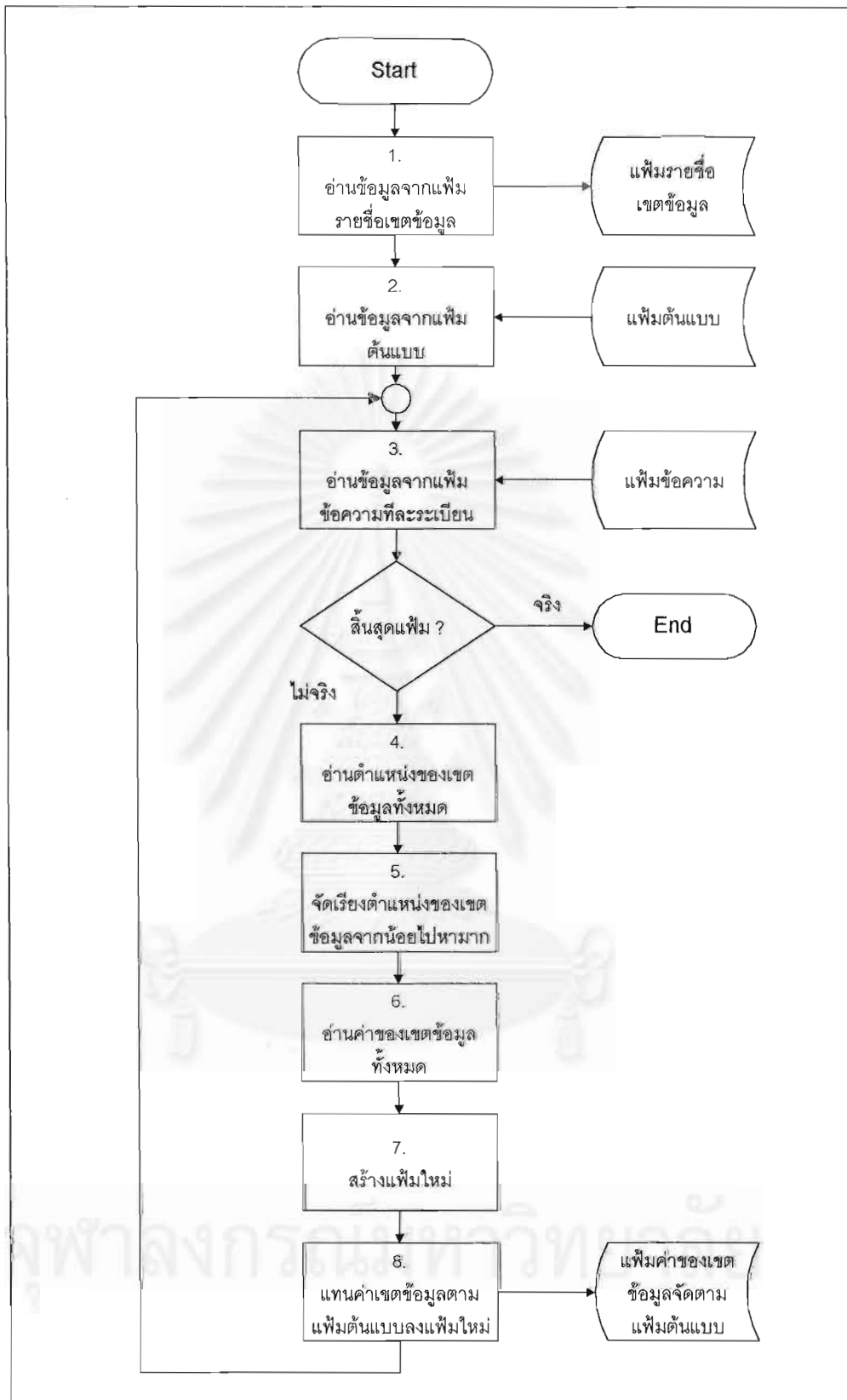
1. อ่านข้อมูลในแฟ้มรายชื่อเขตข้อมูล บันทึกในหน่วยความจำ
2. อ่านข้อมูลในแฟ้มต้นแบบ บันทึกในหน่วยความจำ
3. อ่านข้อมูลจากแฟ้มข้อความที่บันทึกข้อมูลโครงสร้างอธิบายตนเองครั้งละ 1 ระเบียบ
4. อ่านค่าตำแหน่งเริ่มต้นของชื่อเขตข้อมูลในระเบียบ ทำจนกระทั่งครบทุกเขตข้อมูล โดยกำหนดให้เขตข้อมูลสุดท้ายคือตัวคั่นระหว่างระเบียบ โดยค่านิ่งว่า ต้องหาตำแหน่งของกลุ่มของตัวอักษรที่มีชื่อเขตข้อมูลและตัวคั่นเขตข้อมูลเรียงต่อกัน ทั้งนี้เพื่อป้องกันว่า อาจมีการบันทึกชื่อของเขตข้อมูลลงในค่าของเขตข้อมูลก็ได้
5. จัดเรียงตำแหน่งของแต่ละเขตข้อมูลใหม่ โดยเรียงจากตำแหน่งน้อยไปหามาก วิธีนี้เป็น การแก้ปัญหาถ้ากรณีในแฟ้มข้อความเรียงตำแหน่งเขตข้อมูลไม่ตรงกับแฟ้มบันทึกรายชื่อเขตข้อมูล โปรแกรมก็ยังคงอ่านค่าของเขตข้อมูลได้ถูกต้อง
6. อ่านค่าของแต่ละเขตข้อมูลโดยอ่านจากตำแหน่งถัดจากตัวคั่นของเขตข้อมูลปัจจุบัน ไปยังตำแหน่งของชื่อข้อมูลถัดไป -1

ชื่อเขตข้อมูล 1 ค่าของเขตข้อมูล 1 ชื่อเขตข้อมูล 2

NAME: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx AUTHOR: xxxx...



7. สร้างแฟ้มใหม่ตามที่ระบุจากส่วนกำหนดชื่อเพิ่ม (TSCMD.writeto)
8. แทนค่าของแต่ละเขตข้อมูลลงในแต่ละบรรทัดของแฟ้มต้นแบบจนกระทั่งจบแฟ้มต้นแบบ ทำซ้ำตั้งแต่ ข้อ 3 จนกระทั่งจบแฟ้มข้อความ



รูปที่ 4.4 แสดงขั้นตอนการทำงานของโปรแกรมจัดเตรียมเพิ่มข้อมูล

ตัวอย่าง

เพิ่มข้อความรายชื่อหนังสือ ประกอบด้วยระเบียบข้อมูลจำนวนหนึ่ง ในแต่ละระเบียบ ประกอบด้วยรายชื่อเขตข้อมูลดังนี้

1. NAME รายชื่อหนังสือ
2. AUTHOR ผู้แต่งหรือผู้จัดทำ
3. ISBN เลข ISBN ของหนังสือ

รายละเอียดบางส่วนของข้อมูล

NAME:สร้าง Web Page แบบมีอาร์ทด้วย HTML AUTHOR:กิตติ ภัคดีวัฒน์กุล ISBN:974-7042-27-4##

NAME:ศัพท์คอมพิวเตอร์ AUTHOR:ราชบัณฑิตยสถาน ISBN:974-8123-10-3##

NAME:Programming perl AUTHOR:Larry Wall and Randal L. Schwartz ISBN:0-937175-64-1##

...

ขั้นตอนการจัดเตรียมเพิ่มข้อมูล

1. สร้างเพิ่มรายชื่อเขตข้อมูล มีรายละเอียดดังนี้

```
// กำหนดค่าเริ่มต้น
.field_delimiter :
.record_delimiter ##
.counter:8      1
NAME           // ชื่อหนังสือ
AUTHOR         // ผู้แต่ง หรือ ผู้จัดทำ
ISBN
```

2. สร้างเพิ่มเติมแบบสำหรับแสดง มีรายละเอียดดังนี้

```
<html><head>
<meta name="TSCMD.writeto" content="%counter%.htm">
<meta-equiv="content-type" content="text/html">
<title>%NAME%</title></head>
<body>
ชื่อหนังสือ <b>%NAME%</b><br>
ผู้แต่ง/ผู้จัดทำ <b>%AUTHOR%</b><br>
ISBN <b>%ISBN%</b><br>
</body></html>
```

3. สร้างเพิ่มเติมแบบสำหรับสร้างเพิ่มเติมดัชนี มีรายละเอียดดังนี้

```
<html><head>
<meta name="TSCMD.writeto" content="temp\%counter%.htm">
<meta-equiv="content-type" content="text/html">
<meta name="bookname" content="%NAME,t%">
<title>%NAME%</title></head>
<body>
%NAME,t%
%AUTHOR,t%
%ISBN%
</body></html>
```

เมื่อผ่านการประมวลผลจะได้เพิ่มเติมใหม่ สำหรับข้อมูลระเบียบที่ 1 ดังนี้

1. เพิ่ม 00000001.htm สำหรับแสดง

```
<html><head>
<meta-equiv="content-type" content="text/html">
<title>สร้าง Web Page แบบมืออาชีพด้วย HTML </title></head>
<body>
ชื่อหนังสือ <b>สร้าง Web Page แบบมืออาชีพด้วย HTML </b><br>
ผู้แต่ง/ผู้จัดทำ <b>กิตติ ภัคดีวัฒนะกุล</b><br>
ISBN <b>974-7042-27-4</b><br>
</body></html>
```

2. เพิ่ม temp/00000001.htm สำหรับสร้างเพิ่มเติมดัชนี

```
<html><head>
<meta-equiv="content-type" content="text/html">
<meta name="bookname" content="สร้าง Web Page แบบ มือ อาชีพ ด้วย HTML ">
<title>สร้าง Web Page แบบมืออาชีพด้วย HTML </title></head>
<body>
สร้าง Web Page แบบ มือ อาชีพ ด้วย HTML
กิตติ ภัคดี วัฒนะ กุล
974-7042-27-4
</body></html>
```

จะเห็นได้ว่า แฟ้ม temp/00000001.htm จะมีการจัดทำดัชนีคำสำหรับสร้างแฟ้มดัชนี เว้นส่วน
รหัส title ที่ไม่ได้ผ่านขั้นตอนวิธีจัดดัชนีคำ เนื่องจากส่วนนี้กันไว้สำหรับให้โปรแกรมค้นหา
SWISH-E ใช้บันทึกเป็นรายละเอียดประกอบเวลาแสดงผลการค้นหา ในบางโปรแกรมค้นหาจะใช้
ส่วนที่เก็บข้อมูลในรหัส meta tag เป็นรายละเอียดประกอบ ซึ่งสามารถเปลี่ยนแปลงให้สอดคล้อง
ได้โดยเพียงเปลี่ยนแปลงรายละเอียดในแฟ้มต้นแบบเท่านั้น



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5

การพัฒนาระบบสนับสนุนการค้นคืนบทความวิทยานิพนธ์

เมื่อได้พัฒนาโปรแกรมจัดเตรียมเพิ่มข้อมูลในบทที่ 4 ซึ่งสามารถใช้ได้ในกรณีทั่วไปแล้ว ขั้นตอนต่อมาคือ การพัฒนาระบบสนับสนุนการค้นคืนบทความวิทยานิพนธ์ของจุฬาลงกรณ์มหาวิทยาลัย โดยข้อมูลบทความวิทยานิพนธ์จะบันทึกด้วยโครงสร้างแบบอธิบายตนเอง ประกอบด้วยข้อมูลบทความตั้งแต่ ปี พ.ศ. 2538 - 2540 (1 ระเบียบวน ต่อ 1 บทความ) ประกอบด้วยเขตข้อมูลดังนี้

- | | |
|------------------|---|
| 1. TOPIC | หมายถึง ชื่องานวิจัย |
| 2. IDNO | หมายถึง รหัสประจำตัวผู้จัดทำ |
| 3. TNAME | หมายถึง ชื่อ/นามสกุลผู้จัดทำ (ภาษาไทย) |
| 4. ENAME | หมายถึง ชื่อ/นามสกุลผู้จัดทำ (ภาษาอังกฤษ) |
| 5. ADVISOR | หมายถึง ชื่อ/นามสกุลอาจารย์ ที่ปรึกษา |
| 6. DEPARTMENT | หมายถึง คณะ/ภาควิชา |
| 7. YEAR | หมายถึง ปีที่จัดทำ |
| 8. KEYWORD | หมายถึง คำสำคัญ |
| 9. THAI ABSTRACT | หมายถึง บทความย่อ (ภาษาไทย) |
| 10. ENG ABSTRACT | หมายถึง บทความย่อ (ภาษาอังกฤษ) |

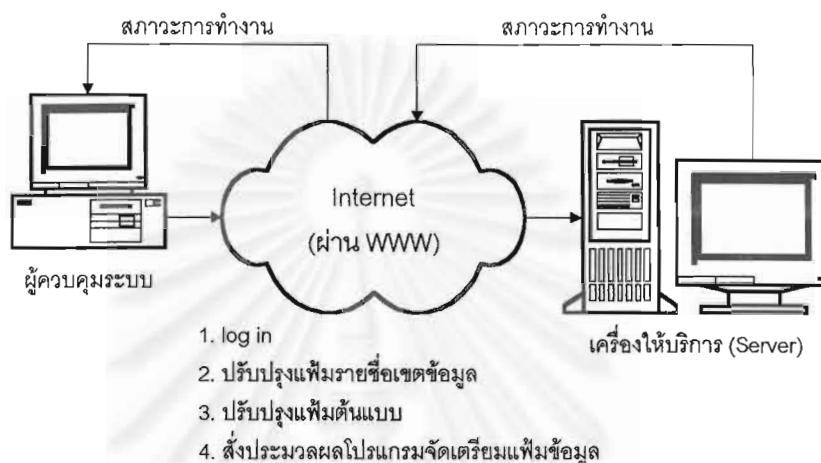
แต่ละเขตข้อมูลจะถูกคั่นด้วย ':' เมื่อสิ้นสุดรายงานแต่ละฉบับ จะถูกคั่นด้วย '~~~~~~'
(ดังตัวอย่างจากภาคผนวก ค.)

ระบบสนับสนุนการค้นคืนบทความวิทยานิพนธ์ ฯ ประกอบด้วย

1. ส่วนจัดเตรียมข้อมูล สำหรับสร้างแฟ้มดัชนีและสร้างแฟ้มสำหรับแสดงรายงาน ซึ่งถูกพัฒนาให้สามารถควบคุมได้จากกระยะไกลผ่านเครือข่าย
2. ส่วนติดต่อกับผู้ใช้ สำหรับให้ผู้ใช้ระบุข้อความ
3. ส่วนปรับปรุงข้อความเพื่อส่งไปประมวลผลที่โปรแกรมค้นหาและรับคืนผลลัพธ์จากโปรแกรมค้นหานำมาจัดรูปแบบและแสดงผลการค้นหา

การพัฒนาส่วนจัดเตรียมข้อมูลแบบควบคุมระยะไกล

การพัฒนาส่วนนี้จะช่วยให้ผู้ควบคุมระบบสามารถจัดเตรียมเพิ่มข้อมูลโดยไม่ต้องมาทำที่เครื่องให้บริการ โดยมีเงื่อนไขคือ เครื่องที่ผู้ควบคุมระบบใช้ กับ เครื่องให้บริการจะต้องอยู่บนเครือข่ายอินเทอร์เน็ต การควบคุมระยะไกลนี้จะทำงานด้วยโปรแกรมบทคำสั่งซีจีไอโดยใช้ภาษาเพิร์ล ผ่านบริการเว็ลด์ไวด์เว็บที่อยู่บนเครือข่ายอินเทอร์เน็ต ดังนั้นผู้ควบคุมระบบสามารถส่งเพิ่มข้อมูล เพิ่มรายชื่อเขตข้อมูล หรือ เพิ่มต้นแบบมายังเครื่องให้บริการโดยผ่านบริการเอฟทีพีก็ได้



รูปที่ 5.1 แสดงการเชื่อมต่อของการจัดเตรียมข้อมูลแบบควบคุมระยะไกล

สิ่งที่ผู้ใช้สามารถทำได้จากการควบคุมระยะไกล

1. ตรวจสอบรหัสผ่าน
2. สร้างหรือแก้ไขเพิ่มรายชื่อเขตข้อมูล
3. สร้างหรือแก้ไขเพิ่มต้นแบบ
4. เริ่มให้โปรแกรมทำการจัดเตรียมเพิ่มข้อมูล โดยระหว่างที่โปรแกรมกำลังประมวลผล

ผู้ควบคุมระบบสามารถตรวจสอบสถานะการทำงานได้

การตรวจสอบรหัสผ่าน

ในการทำงานแต่ละครั้ง ระบบจะอนุญาตให้ผู้ใช้สามารถเรียกทำงานในหัวข้อประมวลผลโปรแกรมจัดเตรียมเพิ่มข้อมูลได้ครั้งละ 1 คนในเวลาเดียวกันเท่านั้น การตรวจสอบการทำงาน ระบบจะทำการสร้างเพิ่มขึ้น 1 เพิ่ม ชื่อ doing.txt ซึ่งเพิ่มนี้จะถูกปิดและลบออกเมื่อสิ้นสุดการประมวลผล ถ้าไม่มีผู้ใช้คนอื่นกำลังเรียกใช้งานอยู่ ระบบจะทำการสร้างฟอร์มของเฮชทีเอ็มแอลเพื่อให้ผู้ใช้ใส่ชื่อและรหัสผ่าน หลังจากนั้นก็ส่งชื่อผู้ใช้และรหัสผ่านกับมายังระบบเพื่อทำการตรวจสอบ ถ้าชื่อผู้ใช้และรหัสผ่านถูกต้อง ระบบจะทำการสร้างเพิ่มผู้ใช้เพื่อบันทึกเวลาทำงาน โดยกำหนดชื่อเพิ่มจากชื่อ

ผู้ใช้ร่วมกับตัวเลขเฉพาะซึ่งได้มาจากการสุ่มหาตัวเลขตั้งแต่ 1 – 100 โดยเลขเฉพาะนี้จะถูกบันทึกใน
ทุกแบบฟอร์มของเฮทที่เอ็มแอลที่ผู้ใช้เรียกใช้งาน ระบบจะมีการตรวจสอบเลขเฉพาะในทุกหัวข้อที่ผู้
ใช้เรียกใช้และเปรียบเทียบเวลาปัจจุบันกับเวลาที่บันทึกการใช้งานล่าสุดในแฟ้มผู้ใช้ว่ามีผลต่างเกิน
กว่าช่วงเวลาที่กำหนดให้รอการเรียกใช้งานระบบหรือไม่ ถ้าเกิน ระบบจะปฏิเสธการเรียกใช้งานออก
จากระบบ ในกรณีนี้ผู้ใช้จะต้องเข้าระบบใหม่เพื่อใส่ชื่อและรหัสผ่าน หลังจากที่ผู้ใช้เรียกใช้งานระบบ
เรียบร้อยแล้วและออกจากระบบ ระบบจะทำการลบแฟ้มผู้ใช้ออก

การสร้างหรือแก้ไขแฟ้มรายชื่อเขตข้อมูล หรือ แฟ้มต้นแบบ

เมื่อผู้ใช้เรียกใช้งานในส่วนนี้ ระบบจะทำการสร้างฟอร์มของเฮทที่เอ็มแอลให้ผู้ใช้ใส่ชื่อแฟ้ม
หลังจากนั้นก็ทำการตรวจสอบก่อนว่ามีแฟ้มนี้หรือไม่ ถ้ามีระบบจะอ่านข้อมูลในแฟ้มเก็บลงตัวแปรที่
ถูกกำหนดให้เป็นแบบ “text field” ถ้าไม่มีระบบจะกำหนดค่าของตัวแปรแบบ “text field” นั้นให้เป็น
ค่าว่าง เมื่อผู้ใช้แก้ไขปรับปรุงเรียบร้อยแล้วส่งรายการมายังระบบ ระบบจะทำการปรับปรุงแฟ้มโดย
ตรวจสอบก่อนว่ามีแฟ้มนี้อยู่หรือไม่ ถ้าไม่มีระบบก็จะบันทึกลงในแฟ้มที่สร้างขึ้นใหม่ ถ้ามีระบบก็จะ
ทำการบันทึกข้อมูลลงในแฟ้มเดิม

การประมวลผลโปรแกรมจัดเตรียมแฟ้มข้อมูล

เมื่อผู้ใช้เรียกใช้งานในส่วนนี้ ระบบจะทำการสร้างเว็บเพจที่มีจุดเชื่อมโยงสำหรับให้ผู้ใช้
สามารถเรียกดูแฟ้มสภาวะการทำงาน (log file)

โปรแกรมที่จะใช้ประมวลผลร่วมประกอบด้วย

1. โปรแกรมจัดเตรียมแฟ้มข้อมูล (gnmain.exe) สำหรับจัดเตรียมแฟ้มสำหรับแสดงราย
งานและแฟ้มสำหรับสร้างแฟ้มดัชนี
2. โปรแกรมค้นหา SWISH-E (swish.exe) สำหรับสร้างและรวมแฟ้มดัชนี

ขั้นตอนการประมวลผลโปรแกรมจัดเตรียมแฟ้มข้อมูล ประกอบด้วย

1. สร้างแฟ้ม doing.txt สำหรับป้องกันไม่ให้ผู้ใช้คนอื่นเรียกซ้ำ
2. จัดเตรียมแฟ้มสำหรับแสดงรายงานบทคัดย่อ
3. จัดเตรียมแฟ้มสำหรับสร้างแฟ้มดัชนี
4. สร้างแฟ้มดัชนีชั่วคราวจากแฟ้มที่ได้จากข้อ 3
5. รวมแฟ้มดัชนีที่สร้างใหม่เข้ากับแฟ้มดัชนีที่มีอยู่เดิม
6. ลบแฟ้มที่สร้างจากข้อ 3
7. ลบแฟ้ม doing.txt

รายละเอียดเพิ่มรายชื่อเขตข้อมูล

```
// ***** field.lst *****
.field_delimiter  ::
.record_delimiter  ~~~~~

TOPIC,N
IDNO,NF
TNAME,N
ENAME,N
ADVISOR,N
DEPARTMENT,N
YEAR,N
KEYWORD
THAI ABSTRACT
ENG ABSTRACT
```

รายละเอียดเพิ่มเติมแบบเพิ่มสำหรับสร้างแฟ้มดัชนี

```
// ***** genidx.htm *****
<html><head>
<meta name="TSCMD.writeto" content=" c:\Thesis\root\ts-bin\doctemp\%YEAR%\%IDNO%.htm">
<meta name="IDNO" content="%IDNO%">
<meta name="TOPIC" content="%TOPIC,T%">
<meta name="AUTHOR" content="%TNAME,T% %ENAME%">
<meta name="ADVISOR" content="%ADVISOR,T%">
<meta name="KEYWORD" content="%KEYWORD,NT%">
<title>%TOPIC%</title></head>
<body>
%TOPIC,NT%
%YEAR%
%TNAME,T%
%ENAME%
%IDNO%
%DEPARTMENT%
%ADVISOR,T%
%KEYWORD,T%
%THAI ABSTRACT,T%
%ENG ABSTRACT%
</body></html>
```

รายละเอียดเพิ่มเติมแบบสำหรับประมวลผลสร้างแฟ้มสำหรับแสดงรายงานบทคัดย่อ ฯ

```
// ***** genhtm.htm *****
.prefix c:\Thesis\root\ts-docs\absdocs\%YEAR%\
.suffix .htm
<html><head>
<meta name="TSCMD.writeto" content=" c:\Thesis\root\ts-docs\absdocs\%YEAR%\%IDNO%.htm">
<title>%TOPIC%</title></head>
<body><basefont face="AngsanaUPC,CordiaUPC,DilleniaUPC"><center>
<table border="1" width="100%" height="224" cellspacing="0">
<tr>
<td width="17%" height="22"><p align="center">ชื่อเรื่อง</td>
<td width="83%" colspan="3" height="22">
<p align="center"><strong>
<font color="#0000FF">%TOPIC%</font></strong></td>
</tr>
<tr>
```


มีปัญหาบางอย่างเกี่ยวกับการจัดทำดัชนีคำ เนื่องจากต้นฉบับของแฟ้มข้อความที่บันทึกข้อมูลบทความย่อ ๆ ส่วนที่เป็นรายละเอียดบทความภาษาไทย มีการขึ้นบรรทัดใหม่ที่ไม่ถูกต้องโดยมีบางส่วนของคำถูกตัดไป ตัวอย่าง

..

และใช้การศึกษาภาคบังคับเป็นเครื่องมือในการกล่อมเกล่าให้มีคุณลักษณะของความเป็นพลเมืองดีคือเป็นคนดี มีความรับผิดชอบ ประกอบอาชีพสุจริต

..

ในการจัดทำดัชนีคำจะใช้วิธีส่งข้อมูลไปฟังก์ชันจัดทำดัชนีคำที่ละบรรทัด เพื่อลดปัญหาข้างต้นจึงได้นำคำสั่งท้ายของบรรทัดที่ผ่านการจัดทำดัชนีคำเรียบร้อยแล้ว มาไว้หน้าบรรทัดใหม่แล้วส่งไปฟังก์ชันจัดทำดัชนีคำอีกครั้ง

จากตัวอย่างข้างต้น จะสามารถแสดงการทำงานได้ดังนี้

ครั้งที่ 1

thaiword(buffer,"และใช้การศึกษาภาคบังคับเป็นเครื่องมือในการกล่อมเกล่าให้มีคุณลักษณะของค ");
คำสั่งท้ายที่อยู่ใน buffer จะได้ "ของค"

ครั้งที่ 2

thaiword(buffer,"ของความเป็นพลเมืองดีคือเป็นคนดี มีความรับผิดชอบ ประกอบอาชีพสุจริต ");

การสร้างแฟ้มดัชนี

ระบุคำสั่ง

```
swish -i c:\thesis\root\ts-bin\doctemp\*.htm -c swish.cfg -f thesis.tmp
```

การรวมแฟ้มดัชนี

ระบุคำสั่ง

```
swish -M thesis.tmp thesis.idx
```

การพัฒนาส่วนติดต่อกับผู้ใช้เพื่อรับข้อความ

ส่วนติดต่อกับผู้ใช้จะเป็นส่วนที่ให้ผู้ใช้งานป้อนข้อความที่ต้องการค้นหา จะถูกสร้างบนเว็บเพจ โดยอาศัยรูปแบบฟอร์มของเฮททีเอ็มแอล โดยกำหนดวิธีการร้องขอแบบ "Get" โดยระบุ

```
<FORM ACTION="/ts-bin/search.pl" METHOD="Get">
```

ส่วนที่ให้ผู้ใช้งานป้อนข้อมูล (ดูรูปที่ 5.2) ประกอบด้วย

1. ตัวแปรในรูปแบบของ "Text Box" สำหรับให้ผู้ใช้งานป้อนข้อความที่ต้องการค้นหา

2. ตัวแปรในรูปแบบของ "Checkbox" สำหรับให้ผู้ใช้เลือกขอบเขตในการค้นหา ประกอบด้วย หัวเรื่อง , ชื่อผู้แต่ง , อาจารย์ที่ปรึกษา , คำสำคัญ ทั้ง 4 หัวข้อ สามารถเลือกได้มากกว่า 1 หัวข้อ
3. ปุ่ม "ค้นหา" สำหรับให้ผู้ใช้กดเมื่อผู้ใช้ต้องการให้เริ่มค้นหา
4. ตัวแปรในรูปแบบ "Drop down list" สำหรับให้ผู้ใช้เลือกที่สามารถแสดงรายชื่อ บทความที่ย่อยที่ค้นคืนได้สูงสุดหน้าละกี่รายการ (10 , 15 , 20 , 25) ค่าโดยปริยาย 10



รูปที่ 5.2 แสดงหน้าจอส่วนที่รับข้อความที่ต้องการค้นหาจากผู้ใช้งาน

การพัฒนาส่วนค้นหาและแสดงผล

รายการต่างๆ ที่ผู้ใช้ป้อนเพื่อค้นหาจะถูกส่งตามมาตรฐานซีจีไอ มายังโปรแกรม search.pl ที่เขียนขึ้นโดยภาษาเพิร์ล มีขั้นตอนการทำงานดังนี้ (ดูรูปที่ 5.3)

1. ถอดรหัสและแปลงข้อความให้เป็นรหัสแอสกี
2. ปรับปรุงข้อความโดยเรียกโปรแกรมจัดทำดัชนีคำ (thaiword.exe) และรับคืนผลลัพธ์จากอุปกรณ์ส่งออกมาตรฐาน
3. ตรวจสอบขอบเขตการค้นหา

ถ้ามีให้เพิ่ม Meta tag ตามขอบเขตที่ระบุ

ตัวอย่าง

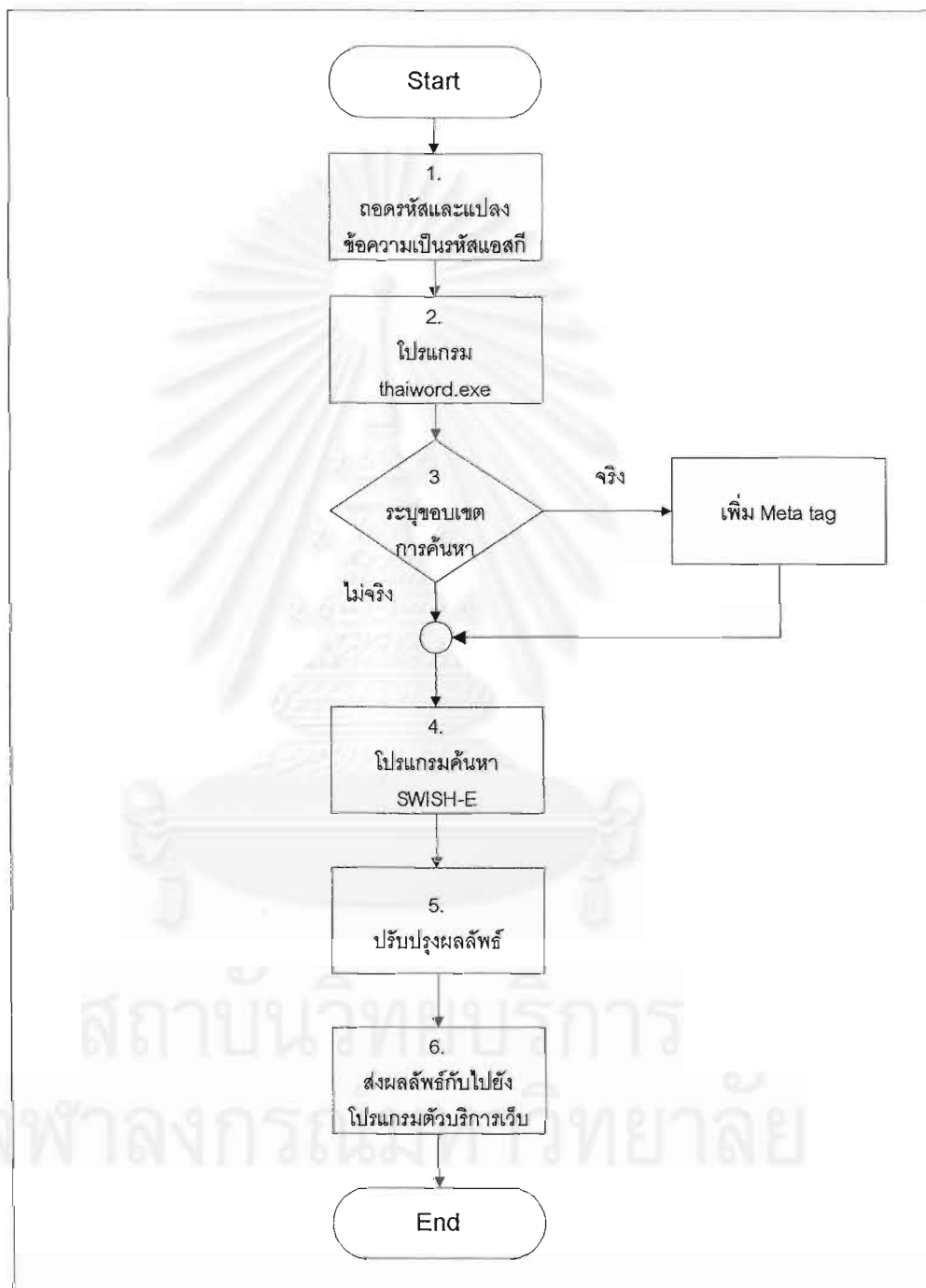
ค้นหาคำว่า "คอมพิวเตอร์" โดยระบุให้ค้นหาเฉพาะหัวเรื่องและคำสำคัญเท่านั้น

โปรแกรมจะเพิ่ม Meta tag ดังนี้

ข้อความใหม่: TOPIC="คอมพิวเตอร์ " or KEYWORD="คอมพิวเตอร์ "

4. ส่งข้อความไปยังโปรแกรมค้นหา SWISH-E และรับผลลัพธ์ทางหน่วยส่งออกมาตรฐาน รายละเอียดผลลัพธ์ที่ได้จากโปรแกรมค้นหา SWISH-E กรณีค้นหาพบ ประกอบด้วย
 1. ชื่อแฟ้มและสถานที่เก็บ
 2. ข้อความส่วนที่ระบุอยู่ในรหัส <TITLE> ... </TITLE>
 3. น้ำหนักของข้อความที่อยู่ในเอกสาร (Rank)

5. ปรับปรุงผลลัพธ์นี้โดยการเพิ่มรหัสเลขที่เอ็มแอลและจุดเชื่อมโยงสถานที่เก็บแฟ้ม ให้ชี้ไปยังสถานที่เก็บแฟ้มเพื่อแสดง
6. ส่งผลลัพธ์กลับไปยังตัวบริการเว็บเพื่อส่งไปยังผู้ใช้ต่อไป



รูปที่ 5.3 แสดงขั้นตอนการทำงานของโปรแกรม Search.pl

รายละเอียดของโปรแกรม Search.pl สามารถดูได้จากภาคผนวก ข

ผลการค้นหา "คอมพิวเตอร์" พบ 150 รายการ

[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15] [ต่อไป]

| | | |
|---|--------------------------|--|
| 1 | B2823378 | A development of software for analysis and design of linear control systems on a microcomputer / Supachai Vorapojpisut |
| 2 | B2806514 | Development of a computer system for sports / Pairat Chaichanadee |
| 3 | B2811133 | Development of a computer system for sports / Pairat Chaichanadee |
| 4 | B2823366 | A computer aided instruction system for computer-aided production management on inventory control. / Pairat Chaichanadee |

รูปที่ 5.4 ตัวอย่าง เว็บเพจที่แสดงผลลัพท์การค้นหา

[กลับ]

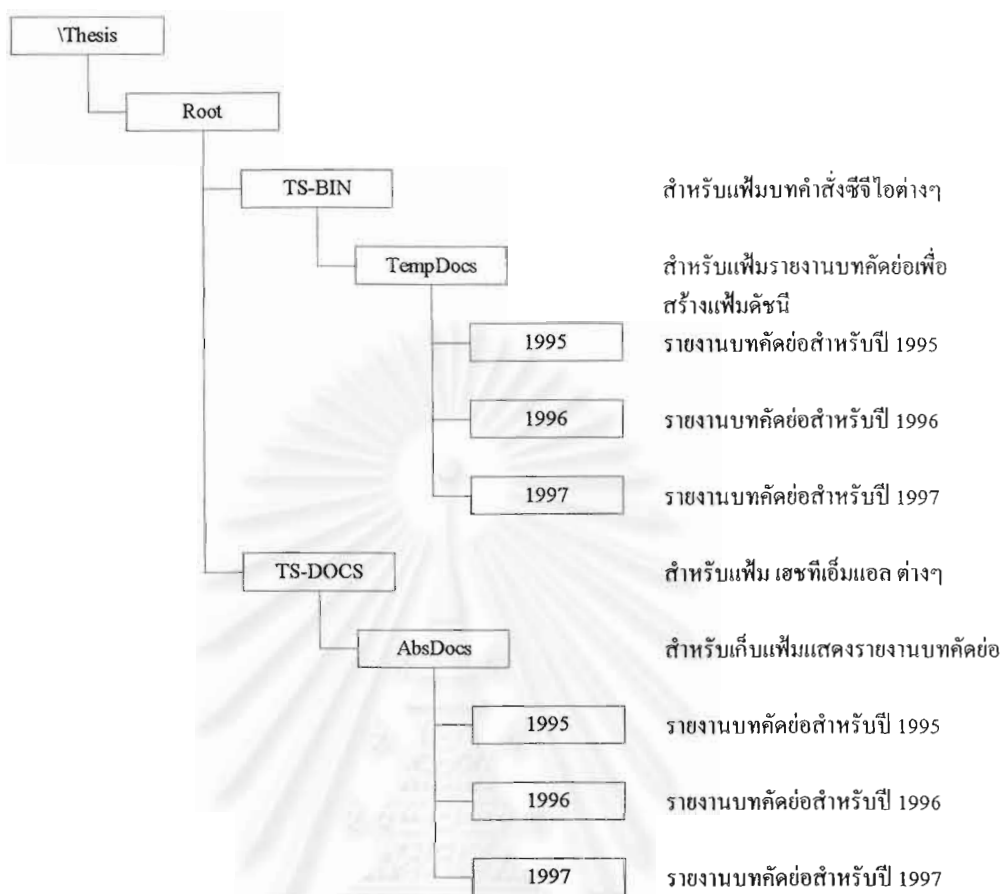
| | | | |
|--------------------|--|------------|------|
| ชื่อเรื่อง | A development of software for analysis and design of linear control systems on a microcomputer / Supachai Vorapojpisut | | |
| จัดทำโดย | ศุภชัย วรพจน์พิศุทธิ์ Supachai Vorapojpisut | | |
| เลขประจำตัว | B2823378 | ปีที่จัดทำ | 1995 |
| อาจารย์ที่ปรึกษา | Varaporn Jaovisidha | | |
| คำสำคัญ | System design Computer programs System analysis Computer programs Linear control systems | | |
| บทคัดย่อภาษาไทย | วิทยานิพนธ์ฉบับนี้ นำเสนอขั้นตอนและรายละเอียดของการพัฒนาระบบคอมพิวเตอร์ สำหรับวิเคราะห์และออกแบบระบบควบคุมเชิงเส้น การคำนวณภายในซอฟต์แวร์อิงบนทฤษฎีควบคุม 3 แบบ คือ ทฤษฎีแบบคลาสสิก แบบสมัยใหม่ และแบบดิจิทัล ในการวิเคราะห์และออกแบบระบบเชิงเส้นที่ไม่ขึ้นกับเวลา โปรแกรมพัฒนาขึ้นด้วยภาษา C++ และทำงานภายใต้โปรแกรม Microsoft Windows ตั้งแต่รุ่น 3.1 ขึ้นไป | | |
| บทคัดย่อภาษาอังกฤษ | This research presents a computation program for analysis and design linear time-invariant control systems based on classical, modern and digital control theory. Software is developed by C++ language and run under Microsoft Windows version 3.1 environment. | | |

รูปที่ 5.5 ตัวอย่าง เว็บเพจแสดงรายงานบทคัดย่อ

จุฬาลงกรณ์มหาวิทยาลัย

การติดตั้งระบบสำหรับเครื่องให้บริการ

โครงสร้างของระบบเพิ่มสำหรับระบบสนับสนุนการค้นคืนฯ ประกอบด้วย



รูปที่ 5.6 แสดงโครงสร้างของระบบเพิ่ม

โดย \Thesis จะตั้งสร้างในสารบบหลักของหน่วยขับ (Drive) ในกรณีทีหน่วยขับไม่ใช่ C ต้องปรับปรุงดังนี้ (สมมติ เป็นหน่วยขับ D)

1. แก้ไขบรรทัดแรกของทุกแฟ้มที่มีส่วนขยาย .pl ที่เรียก แฟ้มห้องสมุด (library file)

```
require "C:\\thesis\\root\\ts-bin\\mylib.pl";
```

โดยปรับจาก C: ให้เป็น D:

2. เพิ่ม mylib.pl แก้ไขส่วนที่ตั้งค่า \$baseDIR = C:\\thesis\\root\\ จาก C: เป็น D: หรือตั้งค่าตัวแปรระบบ set TS-HOME = D:\\thesis\\root\

การติดตั้งสำหรับตัวบริการเว็บ

สร้างสารบบเสมือน (Virtual Directory) ดังนี้

```
/ts-docs = c:\\thesis\\root\\ts-docs
```

กำหนดให้แสดงเว็บเพจ

```
/ts-bin = c:\\thesis\\root\\ts-bin
```

กำหนดเป็น "Shell CGI directory"

บทที่ 6

การวิเคราะห์ผลการทำงานและข้อเสนอแนะ

การวัดผลการทำงานของระบบสนับสนุนการค้นคืนบทความวิทยานิพนธ์ระดับมหาบัณฑิตของจุฬาลงกรณ์มหาวิทยาลัย จะแบ่งออกเป็น 3 ประเภท ดังนี้

1. การเตรียมข้อมูลเพื่อสนับสนุนการค้นคืน
2. การค้นคืนและแสดงผลลัพธ์
3. การปรับปรุงโปรแกรมค้นหา SWISH-E

การทดสอบการเตรียมข้อมูลเพื่อสนับสนุนการค้นคืน

เป็นการทดสอบการใช้เนื้อที่หน่วยความจำถาวรสำหรับการเตรียมเพิ่มข้อมูลสำหรับขนาดเพิ่มข้อความที่ต่างกัน

ขอบเขตการทดสอบ

1. เพิ่มข้อความที่ใช้ทดสอบ เป็นเพิ่มข้อความที่บันทึกรายงานบทความวิทยานิพนธ์ของจุฬาลงกรณ์มหาวิทยาลัย โครงสร้างแบบอธิบายตนเอง มีทั้งสิ้น 6 แฟ้ม ขนาด 5,370 กิโลไบต์ 16,110 กิโลไบต์ 26,850 กิโลไบต์ 37,590 กิโลไบต์ 48,330 กิโลไบต์ และ 64,440 กิโลไบต์ ตามลำดับ
2. เพิ่มต้นแบบที่ใช้ทดสอบได้มาจากการเตรียมจากบทที่ 5

ขั้นตอนในการทดสอบ

1. จัดเตรียมแฟ้มสำหรับสร้างแฟ้มดัชนี โดยใช้โปรแกรมจัดเตรียมเพิ่มข้อมูลและกำหนดให้จัดทำดัชนีคำสำหรับข้อมูลส่วนที่เป็นภาษาไทย
2. สร้างแฟ้มดัชนี โดยใช้โปรแกรมค้นหา SWISH-E
3. จัดเตรียมแฟ้มสำหรับแสดง โดยใช้โปรแกรมจัดเตรียมเพิ่มข้อมูล

ผลการทดสอบ

ตารางที่ 6.1 แสดงเนื้อหาหน่วยความจำถาวรที่ถูกใช้ไปเพื่อจัดเตรียมแฟ้ม ดังนี้

| แบบที่ | ขนาด แฟ้มข้อความ (กิโลไบต์) | จำนวน ระเบียน | เนื้อหาหน่วยความจำถาวรที่ต้องใช้ในการจัดเตรียม (กิโลไบต์) | | | | | | |
|--------|-----------------------------------|------------------|---|-----------|--------------------------|-----------|-----------|---------------------|-----------|
| | | | แฟ้มสำหรับแสดง | | แฟ้มสำหรับสร้างแฟ้มดัชนี | | แฟ้มดัชนี | รวมใช้เนื้อที่เพิ่ม | เพิ่มขึ้น |
| | | | แฟ้มต้นแบบ | ใช้ไป (1) | แฟ้มต้นแบบ | ใช้ไป (2) | | | |
| 1 | 5,370 | 1,658 | 2,164 | 8,465 | 511 | 8,123 | 3,844 | 20,432 | 380.48 |
| 2 | 16,110 | 4,974 | 2,164 | 25,395 | 511 | 24,369 | 10,819 | 60,583 | 376.06 |
| 3 | 26,850 | 8,290 | 2,164 | 42,325 | 511 | 40,615 | 17,778 | 100,718 | 375.11 |
| 4 | 37,590 | 11,606 | 2,164 | 59,255 | 511 | 56,861 | 24,754 | 140,870 | 374.75 |
| 5 | 48,330 | 14,922 | 2,164 | 76,185 | 511 | 73,107 | 31,729 | 181,021 | 374.55 |
| 6 | 64,440 | 19,896 | 2,164 | 101,580 | 511 | 97,476 | 42,284 | 241,340 | 374.52 |

ตารางที่ 6.1 แสดงเนื้อหาที่ใช้ในการจัดเตรียมแฟ้มข้อมูล

จากผลการทดสอบแสดงให้เห็นว่าในการจัดเตรียมแฟ้มข้อความบทคัดย่อวิทยานิพนธ์ข้างต้น จะต้องจัดเตรียมเนื้อที่เพิ่มขึ้นอีกประมาณ 4 เท่าเมื่อเปรียบเทียบกับขนาดแฟ้มข้อความ ขนาดของแฟ้มสำหรับสร้างแฟ้มดัชนี (2) นอกจากจะขึ้นอยู่กับข้อมูลในแฟ้มต้นแบบแล้ว ยังขึ้นอยู่กับจำนวนคำไทยที่ถูกจัดทำดัชนีคำด้วย เนื่องจากทุกคำไทยที่ถูกจัดทำดัชนีคำจะถูกแทรกด้วยช่องว่างซึ่งจะมีผลทำให้ต้องใช้เนื้อที่สำหรับช่องว่างเหล่านี้ด้วย

การทดสอบการค้นคืนและแสดงผลลัพธ์

เป็นการทดสอบประสิทธิภาพการค้นคืนข้อความตามแบบที่ได้จัดเตรียมไว้ล่วงหน้าโดยวัดจากเวลาที่ใช้ในประมวลผลใน 1 คำร้อง การทดสอบจะกระทำที่ความแตกต่างตาม

1. ขนาดแฟ้มที่นำมาใช้จัดเตรียม
2. ข้อคำถามที่ต้องการให้ประมวลผล
3. จำนวนการเชื่อมต่อสำหรับการประมวลผลที่ต้องการให้เกิดขึ้นพร้อมกัน

ขอบเขตการทดสอบ

1. เครื่องบริการสำหรับตัวบริการเว็บ
 - COMPAQ รุ่น PROLINEA
 - CPU: Pentium 120 MHz
 - RAM: 56MB
 - Network Card: 3Com Etherlink III ISA Adapter

- ระบบปฏิบัติการ: MS. Windows NT รุ่น 4.0 (Thai enabled)
กำหนดหน่วยความจำเสมือน (Virtual Memory) 200 MB
- ตัวบริการเว็บ: Netscape Enterprise Server รุ่น 3.0

2. เครื่องรับบริการสำหรับตัวรับบริการเว็บ

- เครื่องที่ 1 COMPAQ รุ่น DESKPRO Pentium 120 MHz
 - RAM: 32MB
 - Network Card: COMPEX RL2000(A) PNP
 - ระบบปฏิบัติการ: MS. Windows 95
 - Protocol,Stack: MS.Windows Sockets Version 1.1
- เครื่องที่ 2 COMPAQ รุ่น PROLINEA Pentium 100 MHz
 - RAM: 32MB
 - Network Card: COMPEX RL2000(A) PNP
 - ระบบปฏิบัติการ: MS. Windows 95
 - Protocol,Stack: MS.Windows Sockets Version 1.1
- เครื่องที่ 3 COMPAQ รุ่น PROLINEA Pentium 100 MHz
 - RAM: 32MB
 - Network Card: 3Com Etherlink III ISA Adapter
 - ระบบปฏิบัติการ: MS. Windows 95
 - Protocol,Stack: MS.Windows Sockets Version 1.1

3. เครื่องข่ายที่ใช้ทดสอบระบบ ใช้เครือข่ายภายในของบริษัท กฤษดามหานคร จำกัด (มหาชน) เชื่อมโยงแบบ Ethernet ขณะทำการทดสอบเครือข่ายอยู่ในสภาพว่าง ไม่มีเครื่องบริการอื่นเปิดใช้งาน

4. โปรแกรมที่ใช้ทดสอบ: โปรแกรม WebBench^[14] รุ่น 3.0 ซึ่งใช้สำหรับวัดประสิทธิภาพการติดต่อสื่อสารระหว่างตัวบริการเว็บและตัวรับบริการเว็บ ประกอบด้วย

4.1 โปรแกรม WebBench 3.0 Controller สำหรับเครื่องบริการ

4.2 โปรแกรม WebBench 3.0 Client สำหรับเครื่องรับบริการ

5. เพิ่มข้อมูลที่ใช้ทดสอบมาจากการทดสอบการจัดเตรียมข้อมูลเพื่อสนับสนุนการค้นคืน

ขั้นตอนในการทดสอบ

1. กำหนดข้อความคำถามที่ใช้ทดสอบ แบ่งออกเป็น 3 ข้อความ คือ
 - 1.1 ข้อคำถามสั้น ค้นคืนแล้วได้ผลลัพธ์น้อย ทดสอบโดยหาคำว่า "ดิลก"
 - 1.2 ข้อคำถามสั้น ค้นคืนแล้วได้ผลลัพธ์มาก ทดสอบโดยหาคำว่า "ศึกษา"
 - 1.3 ข้อคำถามแบบตรรก ค้นคืนแล้วได้ผลลัพธ์น้อย แต่ต้องใช้การประมวลผลที่ซับซ้อนมากขึ้นกว่า 1.1 และ 1.2 ทดสอบโดยหาคำว่า "คอมพิวเตอร์+PC* not UNIX"

| รายการ | ข้อความคำถาม | | |
|--|--------------|------------|------------|
| | แบบที่ 1.1 | แบบที่ 1.2 | แบบที่ 1.3 |
| 1. จำนวนรายงานบทความที่ค้นพบ (หน่วยนับ: ระเบียบ) | | | |
| เพิ่มแบบที่ 1 (ทั้งหมด 1,658 ระเบียบ) | 2 | 1,172 | 19 |
| เพิ่มแบบที่ 2 (ทั้งหมด 4,974 ระเบียบ) | 6 | 3,516 | 57 |
| เพิ่มแบบที่ 3 (ทั้งหมด 8,290 ระเบียบ) | 10 | 5,860 | 95 |
| เพิ่มแบบที่ 4 (ทั้งหมด 11,606 ระเบียบ) | 14 | 8,204 | 133 |
| เพิ่มแบบที่ 5 (ทั้งหมด 14,922 ระเบียบ) | 18 | 11,548 | 171 |
| เพิ่มแบบที่ 6 (ทั้งหมด 19,896 ระเบียบ) | 24 | 11,385 | 219 |
| 2. จำนวนข้อมูลที่ได้รับเฉลี่ยในแต่ละคำร้อง (ไบต์) (กำหนดให้แสดงผลการค้นคืนหน้าละ 10 ระเบียบ (แสดงเฉพาะรายชื่อ)) | | | |
| เพิ่มแบบที่ 1 | 1,468 | 14,005 | 5,735 |
| เพิ่มแบบที่ 2 | 3,506 | 30,782 | 6,090 |
| เพิ่มแบบที่ 3 | 5,558 | 46,765 | 6,425 |
| เพิ่มแบบที่ 4 | 5,733 | 64,061 | 6,811 |
| เพิ่มแบบที่ 5 | 5,733 | 80,787 | 7,189 |
| เพิ่มแบบที่ 6 | 5,877 | 104,001 | 7,539 |

ตารางที่ 6.2 แสดงการจัดเตรียมข้อความคำถามสำหรับการทดสอบการค้นคืนและแสดงผลลัพธ์

2. สร้างเพิ่ม Workload จำนวน 3 เพิ่มมีรายละเอียดดังนี้

- เพิ่มที่ 1 (ts_type1.wl) สำหรับข้อความแบบที่ 1.1

```

DEFINE_CLASSES
  CLASS_THESIS: 100
DEFINE_REQUESTS
  CLASS_THESIS:
    GET /ts-bin/search.pl? sf=%B4%D4%C5%A1&np=10
  
```

- แฟ้มที่ 2 (ts_type2.wl) สำหรับข้อความแบบที่ 1.2

```
DEFINE_CLASSES
CLASS_THESIS: 100
DEFINE_REQUESTS
CLASS_THESIS:
GET /ts-bin/search.pl? sf=%C8%D6%A1%C9%D2&np=10
```

- แฟ้มที่ 3 (ts_type3.wl) สำหรับข้อความแบบที่ 1.3

```
DEFINE_CLASSES
CLASS_THESIS: 100
DEFINE_REQUESTS
CLASS_THESIS:
GET /ts-bin/search.pl? sf=%A4%CD%C1%BE%D4%C7%E0
%B5%CD%C3%EC%2BPC*+not+UNIX&np=10
```

3. สร้างแฟ้ม Test Suite เพื่อกำหนดรายละเอียดในชุดทดสอบ (Mix) ดังนี้

ทดสอบที่ 1-3 เครื่องรับบริการ และจากข้อจำกัดที่เครื่องบริการสามารถจัดการ คำร้องขอที่เห็นลที่เชื่อถือได้ ไม่ควรเกิน 15 คำร้องในเวลาเดียวกัน ดังนั้นจึงกำหนดให้ ทดสอบที่จำนวนสายโยงใย (Thread) ตั้งแต่ 1 และ 5 รวม 6 ชุดทดสอบ โดยทุกครั้ง ของการทดสอบจะกำหนดเงื่อนไขอื่นๆ ดังนี้

| ชื่อ | กำหนดค่า | รายละเอียด |
|---------------------------|-------------|---|
| RampUp | 0 | กำหนดเวลาเริ่ม (0 วินาที) |
| RampDown | 0 | กำหนดเวลาก่อนสิ้นสุด (0 วินาที) |
| Length | 300 | กำหนดเวลาที่ใช้ทดสอบ (300 วินาที) |
| Delay | 0 | เวลาที่ให้เครื่องบริการรอกก่อนเริ่มทดสอบ (0 วินาที) |
| Think | 0 | เวลาที่ให้เครื่องบริการรอกก่อนที่จะส่งคำร้องถัดไปหลังจากได้รับการตอบสนอง (0 วินาที) |
| Thread Inactivity Timeout | 180 | เวลาสูงสุดที่จะรอกการตอบสนองจากเครื่องบริการ (180 วินาที) |
| Receive Buffer Size | 4096 | ขนาดข้อมูลสูงสุดที่ตัวรับบริการเว็บสามารถอ่านจากเครื่องบริการใน 1 ครั้ง (4096 ไบต์) |
| WWW Server | Ntstk | ชื่อเครื่องสำหรับตัวบริการเว็บ |
| WWW Port | 80 | ช่องทางของ WWW |
| Workload file | | ชื่อแฟ้ม Workload |
| | Ts_type1.wl | สำหรับทดสอบโดยข้อความแบบที่ 1.1 |
| | Ts_type2.wl | สำหรับทดสอบโดยข้อความแบบที่ 1.2 |
| | Ts_type3.wl | สำหรับทดสอบโดยข้อความแบบที่ 1.3 |

ตารางที่ 6.3 แสดงรายละเอียดเงื่อนไขที่ระบุในชุดทดสอบของโปรแกรม WebBench

เวลาที่ใช้ในการทดสอบ

- 1 ชุดทดสอบจะใช้เวลารวม 300 นาที
- 1 ข้อคำถามจะต้องใช้ 6 ชุดทดสอบ
- การทดสอบเพิ่มทั้ง 6 แบบจะใช้เวลาทดสอบเพิ่มละ 3 ข้อคำถาม
- รวมเวลาที่ใช้ในการทดสอบทั้งหมด $(300 \times 6) \times 6 \times 3 = 32,400$ วินาที

ในการบันทึกผลการทดสอบจะคิดเฉพาะคำร้องที่สมบูรณ์เท่านั้น คำร้องที่สมบูรณ์จะต้องประกอบด้วยขั้นตอนดังต่อไปนี้

1. ตัวรับบริการเว็บติดต่อไปยังตัวบริการเว็บ
2. ตัวรับบริการเว็บส่งคำร้องไปยังตัวบริการเว็บ
3. ตัวบริการเว็บส่งคำร้องไปยังโปรแกรมบทคำสั่งซีจีไอ
4. โปรแกรมบทคำสั่งซีจีไอประมวลผลและส่งผลลัพธ์กลับคืนไปยังตัวบริการเว็บ
5. ตัวบริการเว็บส่งผลลัพธ์กลับไปยังตัวรับบริการเว็บ
6. ตัวรับบริการเว็บตัดการติดต่อกับตัวบริการเว็บ

ผลการทดสอบ

1. แสดงผลการทดสอบสำหรับขนาดเพิ่มข้อความแบบที่ 1 (5,370 กิโลไบต์)

| การเชื่อมต่อ | | | ข้อคำถาม (หน่วยเวลาเป็นวินาที) | | | | | |
|-------------------------|--------------|----------|--------------------------------|-------------|------------|-------------|------------|-------------|
| จำนวน | เว็บไคลเอนต์ | สายโยงใย | แบบที่ 1.1 | | แบบที่ 1.2 | | แบบที่ 1.3 | |
| | | | คำร้อง | เวลา/คำร้อง | คำร้อง | เวลา/คำร้อง | คำร้อง | เวลา/คำร้อง |
| 1 | 1 | 1 | 63 | 4.76 | 42 | 7.14 | 55 | 5.45 |
| 2 | 2 | 1 | 85 | 3.53 | 53 | 5.66 | 72 | 4.17 |
| 3 | 3 | 1 | 96 | 3.13 | 62 | 4.84 | 75 | 4.00 |
| 5 | 1 | 5 | 88 | 3.41 | 44 | 6.82 | 69 | 4.35 |
| 10 | 2 | 5 | 70 | 4.29 | 36 | 8.33 | 60 | 5.00 |
| 15 | 1 | 15 | 78 | 3.85 | 39 | 7.69 | 65 | 4.62 |
| 15 | 3 | 5 | 92 | 3.26 | 42 | 7.14 | 75 | 4.00 |
| เวลาที่ใช้เฉลี่ย/คำร้อง | | | | 2.91 | | 5.29 | | 3.51 |

ตารางที่ 6.4 แสดงผลการทดสอบสำหรับขนาดเพิ่มข้อความแบบที่ 1

2. แสดงผลการทดสอบสำหรับขนาดเพิ่มข้อความแบบที่ 2 (16,110 กิโลไบต์)

| การเชื่อมต่อ | | | ข้อคำถาม (หน่วยเวลาเป็นวินาที) | | | | | |
|-------------------------|-----------------|----------|--------------------------------|-------------|------------|-------------|------------|-------------|
| จำนวน | เว็บเบราว์เซอร์ | สายโยงใย | แบบที่ 1.1 | | แบบที่ 1.2 | | แบบที่ 1.3 | |
| | | | คำร้อง | เวลา/คำร้อง | คำร้อง | เวลา/คำร้อง | คำร้อง | เวลา/คำร้อง |
| 1 | 1 | 1 | 62 | 4.84 | 38 | 7.89 | 53 | 5.66 |
| 2 | 2 | 1 | 78 | 3.85 | 41 | 7.32 | 66 | 4.55 |
| 3 | 3 | 1 | 80 | 3.75 | 40 | 7.50 | 75 | 4.00 |
| 5 | 1 | 5 | 80 | 3.75 | 38 | 7.89 | 66 | 4.55 |
| 10 | 2 | 5 | 63 | 4.76 | 32 | 9.38 | 63 | 4.76 |
| 15 | 1 | 15 | 66 | 4.55 | 32 | 9.38 | 58 | 5.17 |
| 15 | 3 | 5 | 80 | 3.75 | 35 | 8.57 | 66 | 4.55 |
| เวลาที่ใช้เฉลี่ย/คำร้อง | | | | 3.25 | | 6.44 | | 3.69 |

ตารางที่ 6.5 แสดงผลการทดสอบสำหรับขนาดเพิ่มข้อความแบบที่ 2

3. แสดงผลการทดสอบสำหรับขนาดเพิ่มข้อความแบบที่ 3 (26,850 กิโลไบต์)

| การเชื่อมต่อ | | | ข้อคำถาม (หน่วยเวลาเป็นวินาที) | | | | | |
|-------------------------|-----------------|----------|--------------------------------|-------------|------------|-------------|------------|-------------|
| จำนวน | เว็บเบราว์เซอร์ | สายโยงใย | แบบที่ 1.1 | | แบบที่ 1.2 | | แบบที่ 1.3 | |
| | | | คำร้อง | เวลา/คำร้อง | คำร้อง | เวลา/คำร้อง | คำร้อง | เวลา/คำร้อง |
| 1 | 1 | 1 | 57 | 5.26 | 34 | 8.82 | 49 | 6.12 |
| 2 | 2 | 1 | 65 | 4.62 | 38 | 7.89 | 57 | 5.26 |
| 3 | 3 | 1 | 70 | 4.29 | 45 | 6.67 | 66 | 4.55 |
| 5 | 1 | 5 | 65 | 4.62 | 27 | 11.11 | 42 | 7.14 |
| 10 | 2 | 5 | 63 | 4.76 | 25 | 12.00 | 42 | 7.14 |
| 15 | 1 | 15 | 66 | 4.55 | 30 | 10.00 | 43 | 6.98 |
| 15 | 3 | 5 | 73 | 4.11 | 20 | 15.00 | 66 | 4.55 |
| เวลาที่ใช้เฉลี่ย/คำร้อง | | | | 3.58 | | 7.94 | | 4.64 |

ตารางที่ 6.6 แสดงผลการทดสอบสำหรับขนาดเพิ่มข้อความแบบที่ 3

4. แสดงผลการทดสอบสำหรับขนาดเพิ่มข้อความแบบที่ 4 (37,590 กิโลไบต์)

| การเชื่อมต่อ | | | ข้อคำถาม (หน่วยเวลาเป็นวินาที) | | | | | |
|-------------------------|--------------|----------|--------------------------------|-------------|------------|-------------|------------|-------------|
| จำนวน | เว็บไคลเอนต์ | สายใยโยง | แบบที่ 1.1 | | แบบที่ 1.2 | | แบบที่ 1.3 | |
| | | | คำร้อง | เวลา/คำร้อง | คำร้อง | เวลา/คำร้อง | คำร้อง | เวลา/คำร้อง |
| 1 | 1 | 1 | 53 | 5.66 | 27 | 11.11 | 45 | 6.67 |
| 2 | 2 | 1 | 60 | 5.00 | 26 | 11.54 | 49 | 6.12 |
| 3 | 3 | 1 | 65 | 4.62 | 24 | 12.50 | 52 | 5.77 |
| 5 | 1 | 5 | 60 | 5.00 | 24 | 12.50 | 40 | 7.50 |
| 10 | 2 | 5 | 55 | 5.45 | 20 | 15.00 | 41 | 7.32 |
| 15 | 1 | 15 | 55 | 5.45 | 22 | 13.64 | 40 | 7.50 |
| 15 | 3 | 5 | 60 | 5.00 | 20 | 15.00 | 42 | 7.14 |
| เวลาที่ใช้เฉลี่ย/คำร้อง | | | 4.02 | | 10.14 | | 5.34 | |

ตารางที่ 6.7 แสดงผลการทดสอบสำหรับขนาดเพิ่มข้อความแบบที่ 4

5. แสดงผลการทดสอบสำหรับขนาดเพิ่มข้อความแบบที่ 5 (48,330 กิโลไบต์)

| การเชื่อมต่อ | | | ข้อคำถาม (หน่วยเวลาเป็นวินาที) | | | | | |
|-------------------------|--------------|----------|--------------------------------|-------------|------------|-------------|------------|-------------|
| จำนวน | เว็บไคลเอนต์ | สายใยโยง | แบบที่ 1.1 | | แบบที่ 1.2 | | แบบที่ 1.3 | |
| | | | คำร้อง | เวลา/คำร้อง | คำร้อง | เวลา/คำร้อง | คำร้อง | เวลา/คำร้อง |
| 1 | 1 | 1 | 48 | 6.25 | 20 | 15.00 | 38 | 7.89 |
| 2 | 2 | 1 | 59 | 5.08 | 22 | 13.64 | 48 | 6.25 |
| 3 | 3 | 1 | 58 | 5.17 | 22 | 13.64 | 53 | 5.66 |
| 5 | 1 | 5 | 55 | 5.45 | 21 | 14.29 | 38 | 7.89 |
| 10 | 2 | 5 | 55 | 5.45 | 22 | 13.64 | 37 | 8.11 |
| 15 | 1 | 15 | 48 | 6.25 | 20 | 15.00 | 39 | 7.69 |
| 15 | 3 | 5 | 55 | 5.45 | 19 | 15.79 | 35 | 8.57 |
| เวลาที่ใช้เฉลี่ย/คำร้อง | | | 4.35 | | 11.22 | | 5.79 | |

ตารางที่ 6.8 แสดงผลการทดสอบสำหรับขนาดเพิ่มข้อความแบบที่ 5

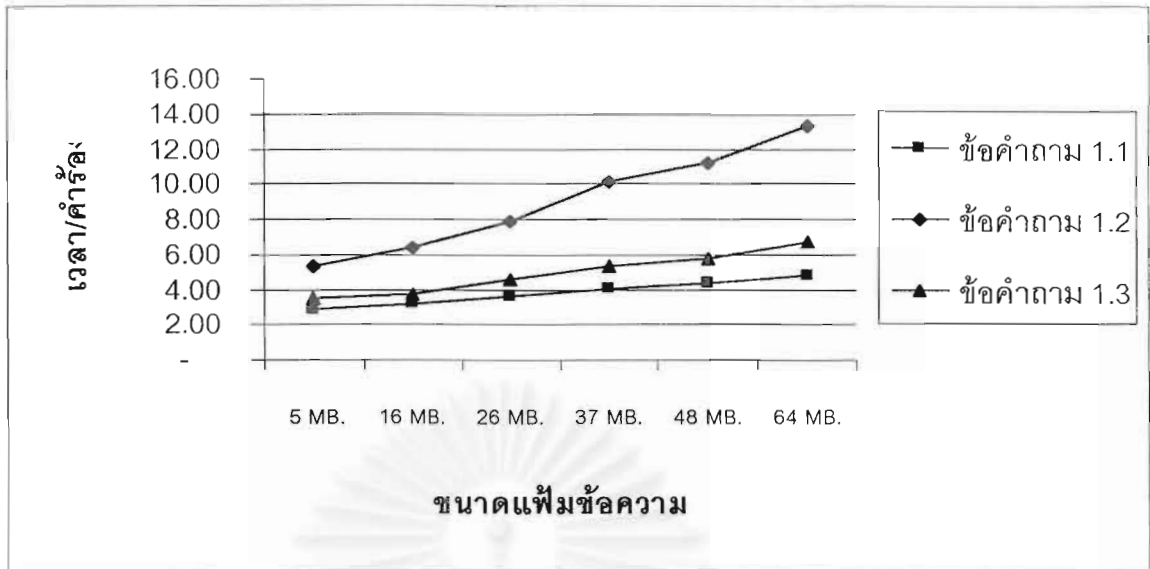
6. แสดงผลการทดสอบสำหรับขนาดเพิ่มข้อความแบบที่ 6 (64,440 กิโลไบต์)

| การเชื่อมต่อ | | | ข้อคำถาม (หน่วยเวลาเป็นวินาที) | | | | | |
|-------------------------|-------------|----------|--------------------------------|-------------|------------|--------------|------------|-------------|
| จำนวน | เว็บโคเดเนต | สายโยงโย | แบบที่ 1.1 | | แบบที่ 1.2 | | แบบที่ 1.3 | |
| | | | คำร้อง | เวลา/คำร้อง | คำร้อง | เวลา/คำร้อง | คำร้อง | เวลา/คำร้อง |
| 1 | 1 | 1 | 43 | 6.98 | 19 | 15.79 | 33 | 9.09 |
| 2 | 2 | 1 | 55 | 5.45 | 20 | 15.00 | 45 | 6.67 |
| 3 | 3 | 1 | 59 | 5.08 | 18 | 16.67 | 50 | 6.00 |
| 5 | 1 | 5 | 50 | 6.00 | 16 | 18.75 | 36 | 8.33 |
| 10 | 2 | 5 | 42 | 7.14 | 17 | 17.65 | 33 | 9.09 |
| 15 | 1 | 15 | 48 | 6.25 | 17 | 17.65 | 29 | 10.34 |
| 15 | 3 | 5 | 49 | 6.12 | 16 | 18.75 | 27 | 11.11 |
| เวลาที่ใช้เฉลี่ย/คำร้อง | | | | 4.78 | | 13.36 | | 6.74 |

ตารางที่ 6.9 แสดงผลการทดสอบสำหรับขนาดเพิ่มข้อความแบบที่ 6

| เพิ่มข้อความ | ค่าเฉลี่ยเวลาที่ใช้ในการค้นคืนต่อคำร้อง (หน่วย: วินาที) | | |
|---------------------------------|---|------------|------------|
| | แบบที่ 1.1 | แบบที่ 1.2 | แบบที่ 1.3 |
| เพิ่มแบบที่ 1 (5,370 กิโลไบต์) | 2.91 | 5.29 | 3.51 |
| เพิ่มแบบที่ 2 (16,110 กิโลไบต์) | 3.25 | 6.44 | 3.69 |
| เพิ่มแบบที่ 3 (26,850 กิโลไบต์) | 3.58 | 7.94 | 4.64 |
| เพิ่มแบบที่ 4 (37,590 กิโลไบต์) | 4.02 | 10.14 | 5.34 |
| เพิ่มแบบที่ 5 (48,330 กิโลไบต์) | 4.35 | 11.22 | 5.79 |
| เพิ่มแบบที่ 6 (64,440 กิโลไบต์) | 4.78 | 13.36 | 6.74 |
| เวลาที่ใช้ น้อยที่สุด | 2.91 | 5.29 | 3.51 |
| เวลาที่ใช้ มากที่สุด | 4.78 | 13.36 | 6.74 |

ตารางที่ 6.10 แสดงค่าเฉลี่ยเวลาที่ใช้ในการค้นคืนต่อคำร้อง



รูปที่ 6.1 แผนภูมิแสดงการเปลี่ยนแปลงค่าเฉลี่ยเวลาที่ใช้ในการค้นคืนต่อคำร้อง

คำนวณหาค่าความลาดชันเฉลี่ยต่อเมกะไบต์ของแต่ละข้อคำถาม

$$= (\text{ค่าเฉลี่ยเวลามากที่สุด} - \text{ค่าเฉลี่ยเวลาน้อยที่สุด}) / (\text{ขนาดเพิ่มใหญ่ที่สุด} - \text{ขนาดเพิ่มเล็กที่สุด})$$

แทนค่าสำหรับข้อคำถามแต่ละแบบ

$$\text{แบบที่ 1.1} = (4.78 - 2.91) / (64 - 5) = 0.031 \text{ วินาทีต่อเมกะไบต์}$$

$$\text{แบบที่ 1.2} = (13.36 - 5.29) / (64 - 5) = 0.137 \text{ วินาทีต่อเมกะไบต์}$$

$$\text{แบบที่ 1.3} = (6.74 - 3.51) / (64 - 5) = 0.055 \text{ วินาทีต่อเมกะไบต์}$$

$$\text{ความลาดชันเฉลี่ยของทุกข้อคำถาม} = (0.031 + 0.137 + 0.055) / 3 = 0.07 \text{ วินาทีต่อเมกะไบต์}$$

จากผลการทดสอบพบว่า ความสัมพันธ์ของเวลาการประมวลผลต่อคำร้องแปรผันเป็นเชิงเส้นกับขนาดเพิ่มข้อมูลด้วยความลาดชันโดยเฉลี่ยประมาณ 0.07 วินาทีต่อเมกะไบต์

การทดสอบการปรับปรุงโปรแกรมค้นหา SWISH-E

เป็นการทดสอบประสิทธิภาพระหว่างโปรแกรมค้นหา SWISH-E ก่อนปรับปรุงให้ใช้กับตัวอักษรภาษาไทยกับหลังปรับปรุงให้ใช้กับตัวอักษรภาษาไทย

ขอบเขตการทดสอบ

1. เพิ่มข้อมูลที่ใช้ทดสอบ ได้มาจากการทดสอบการจัดเตรียมข้อมูลเพื่อสนับสนุนการค้นคืน โดยใช้เฉพาะเพิ่มข้อมูลแบบที่ 1

2. เครื่องบริการ เครื่องรับบริการ เครือข่ายและโปรแกรมที่ใช้ทดสอบ เช่นเดียวกับการทดสอบการค้นคืนและแสดงผลลัพธ์
3. การทดสอบจะกำหนดข้อความเป็นภาษาอังกฤษ ดังนั้นขณะทำการทดสอบจึงกำหนดให้ไม่ต้องใช้โปรแกรมแบ่งคำเพื่อจัดเตรียมข้อความ

ขั้นตอนในการทดสอบ

1. สร้างเพิ่มดัชนีใหม่โดยใช้โปรแกรมค้นหา SWISH-E ทั้ง 2 แบบ
2. กำหนดข้อความ โดยการให้ค้นหาคำว่า "television"

| | ก่อนปรับปรุง ฯ | หลังปรับปรุง ฯ |
|---|----------------|----------------|
| 1. จำนวนรายงานบทคัดย่อทั้งหมด (ระเบียน) | 1,658 | 1,658 |
| 2. จำนวนรายงานบทคัดย่อที่ค้นพบ (ระเบียน) | 71 | 74 |
| 3. จำนวนข้อมูลที่ได้รับเฉลี่ยในแต่ละคำร้อง (ไบต์) (กำหนดให้แสดงผลการค้นคืนหน้าละ 10 ระเบียน แสดงเฉพาะรายชื่อ) | 6,629 | 6,592 |

ตารางที่ 6.8 แสดงการจัดเตรียมข้อความสำหรับการทดสอบโปรแกรมค้นหา SWISH-E

หมายเหตุ สาเหตุที่ผลลัพธ์แตกต่างกันเนื่องมาจากในบางเอกสารมีตัวสัญลักษณ์อื่นต่อท้ายคำ "television" เช่น television| ซึ่งโปรแกรม SWISH-E ก่อนปรับปรุงฯ ได้ทำการรวมเข้าเป็นคำเดียวกันดังนั้นจึงมีผลทำให้ค้นหาไม่พบ ขณะที่โปรแกรม SWISH-E หลังปรับปรุงฯ ซึ่งได้รับการแก้ไขโดยการปรับปรุงตัวแปรสำหรับตรวจสอบตัวอักษร ให้สามารถประมวลผลตัวอักษรรหัสภาษาไทย แต่ให้เว้นตัวอักษรประเภทสัญลักษณ์ จึงมีผลทำให้สัญลักษณ์บางอย่าง เช่น | ไม่ได้ถูกรวมเข้าเพื่อจัดทำดัชนีด้วย

3. สร้างเพิ่ม Workload (swtest.wl) มีรายละเอียดดังนี้

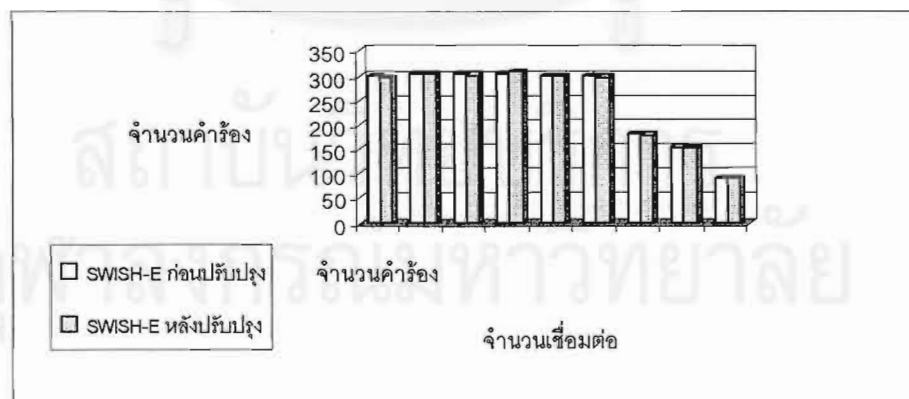
```
DEFINE_CLASSES
  CLASS_THESIS: 100
DEFINE_REQUESTS
  CLASS_THESIS:
    GET /ts-bin/search.pl?sf=television&np=10
```

4. สร้างเพิ่ม Test Suite เพื่อกำหนดรายละเอียดในชุดทดสอบเช่นเดียวกับการทดสอบการค้นคืนและแสดงผลลัพธ์ เว้นส่วนชื่อเพิ่ม Workload ใช้ชื่อ swtest.wl แทน

ผลการทดสอบ

| การเชื่อมต่อ | | | โปรแกรม SWISH-E | | | |
|-----------------------------|----------|----------|-----------------|---------------|----------------|---------------|
| จำนวน | เว็บไซต์ | สายโยงใย | ก่อนปรับปรุง ฯ | | หลังปรับปรุง ฯ | |
| | | | คำร้อง* | เวลา/คำร้อง** | คำร้อง* | เวลา/คำร้อง** |
| 1 | 1 | 1 | 300 | 1.00 | 297 | 1.01 |
| 2 | 2 | 1 | 304 | 0.99 | 304 | 0.99 |
| 3 | 3 | 1 | 304 | 0.99 | 303 | 0.99 |
| 5 | 1 | 5 | 304 | 0.99 | 309 | 0.97 |
| 10 | 2 | 5 | 300 | 1.00 | 300 | 1.00 |
| 15 | 1 | 15 | 301 | 1.00 | 298 | 1.01 |
| 15 | 3 | 5 | 182 | 1.65 | 180 | 1.67 |
| 30 | 2 | 15 | 155 | 1.94 | 153 | 1.96 |
| 45 | 3 | 15 | 90 | 3.33 | 90 | 3.33 |
| เวลาที่ใช้เฉลี่ย/คำร้อง | | | | 1.43 | | 1.44 |
| เวลาที่ใช้น้อยที่สุด/คำร้อง | | | | 0.99 | | 0.97 |
| เวลาที่ใช้มากที่สุด/คำร้อง | | | | 3.33 | | 3.33 |

ตารางที่ 6.8 แสดงผลการทดสอบการปรับปรุงโปรแกรมค้นหา SWISH-E



รูปที่ 6.4 แผนภูมิแสดงผลการทดสอบการปรับปรุงโปรแกรมค้นหา SWISH-E

จากผลการทดสอบแสดงให้เห็นว่า เวลาที่ใช้ในการประมวลในแต่ละคำร้องของทั้งสองโปรแกรมทั้งก่อนและหลังปรับปรุงเพื่อให้สามารถประมวลผลตัวอักษรรหัสภาษาไทย ได้ผลลัพธ์ที่ใกล้เคียงกัน

ข้อเสนอแนะ

จากผลงานวิจัยครั้งนี้ สามารถเป็นแนวคิดให้เกิดงานวิจัยอื่นๆ ได้อีก เช่น

- จากการทดสอบพบว่า โปรแกรม SWISH-E ถูกออกแบบมาให้จัดการข้อมูลในหน่วยความจำชั่วคราว ซึ่งถ้าขนาดเพิ่มข้อมูลใหญ่มากจนระบบปฏิบัติการไม่สามารถจัดสรรหน่วยความจำได้อย่างพอเพียงแล้วจะทำให้เกิดปัญหาขึ้นได้ แนวทางในการแก้ไขปัญหาคือ พัฒนาให้สามารถจัดการข้อมูลในหน่วยความจำถาวรในรูปแบบของหน่วยความจำเสมือน นอกจากนี้ โปรแกรม SWISH-E ประมวลผลคำโดยตรวจสอบจากตัวอักษรคั่นซึ่งกำหนดเฉพาะในโปรแกรม ๆ เช่น ช่องว่าง , จุดภาพ เป็นต้น ดังนั้นถ้าข้อมูลให้อักษรคั่นแทน จะทำให้โปรแกรม SWISH-E ประมวลผลคำได้ไม่ถูกต้อง แนวทางการแก้ปัญหาคือ พัฒนาให้สามารถกำหนดตัวอักษรคั่นได้ในภายหลังโดยไม่ต้องใช้เท่าที่กำหนดเฉพาะในโปรแกรมเท่านั้น
- บทคำสั่งซีไฉ่ที่พัฒนาขึ้นเพื่อสนับสนุนงานวิจัยครั้งนี้ ยังเป็นแบบเริ่มทำงานเมื่อมีคำร้องขอ ซึ่งการเริ่มทำงานแต่ละครั้งต้องใช้เวลาในการอ่านเพิ่มพจนานุกรมและเพิ่มดัชนีใหม่ทุกครั้งทำให้เสียเวลามาก ถ้าสามารถพัฒนาให้เป็นโปรแกรมประเภทฝังตัวและใช้การอ่านเพิ่มพจนานุกรมและเพิ่มดัชนีเพียงครั้งเดียว สามารถประมวลผลต่อเนื่องหลายคำร้องจะทำให้ลดเวลาในการประมวลผลได้มาก
- พัฒนาโปรแกรมค้นหาสำหรับเอกสารไทยโดยใช้ขั้นตอนวิธีจัดทำดัชนีคำ โดยไม่ต้องใช้การจัดเตรียมข้อมูลเพื่อสร้างเพิ่มดัชนีหรือปรับปรุงข้อความก่อน
- ขั้นตอนวิธีสำหรับการค้นหาข้อความสำหรับงานวิจัยนี้ ยังเป็นแบบพื้นฐาน กล่าวคือ ยังต้องให้ผู้ใช้งานกำหนดเงื่อนไขเองว่าต้องการค้นหาแบบตรรกใด เช่น ตรรก AND หรือตรรก OR แนวคิดหนึ่งในงานวิจัยต่อไปก็คือ การวิเคราะห์ข้อความจากผู้ใช้เพื่อกำหนดเครื่องหมายตรรก แล้วค้นหาจากข้อความที่ผ่านการวิเคราะห์นั้น

รายการอ้างอิง

- [1] กิตติ ภัคดีวัฒนะกุล. สร้าง Web Page แบบมีอาชีพด้วย HTML. กรุงเทพมหานคร: บริษัท ดวงกลมสมัย จำกัด, 2540.
- [2] จารุวรรณ ช้วยเดช. การออกแบบและพัฒนาเว็สต์ไซด์เว็บสำหรับการสื่อสารสารสนเทศ. วิทยานิพนธ์ปริญญาโทบริหารธุรกิจ สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2540.
- [3] วิฑูรย์ กัลยาณวัฒน์. ระบบการค้นคืนข้อความภาษาไทยโดยใช้เพิ่มข้อมูลพจนานุกรม. วิทยานิพนธ์ปริญญาโทบริหารธุรกิจ สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2540.
- [4] สุรศักดิ์ สงวนพงษ์. คู่มืออินเทอร์เน็ต. กรุงเทพมหานคร: บริษัท ซีเอ็ดดูเคชั่น จำกัด, 2538.
- [5] อรรถชัย กัมลาศพิทักษ์. รู้จักกับ HyperText Transfer Protocol. วารสาร Internet Magazine 33 (มีนาคม 2542): 61-67.
- [6] Berkeley Digital Library SunSITE. Simple Web Indexing System for Humans-Enhanced (SWISH-E). (URL <http://sunsite.berkeley.edu/SWISH-E/>).
- [7] Innovative Interfaces Inc. (INNOPAC). ฐานข้อมูลวิทยานิพนธ์จุฬาลงกรณ์และดัชนีวารสาร. (URL <http://library.car.chula.ac.th:82/>).
- [8] Larry Wall and Randal L. Schwartz. Programming Perl. O'Reilly & Associates, 1991.
- [9] Michael J. Folk and Bill Zoellick. File Structures. Addison-Wesley, 1992.
- [10] O'Reilly & Associates. Perl Software. (URL <http://www.perl.com/>).
- [11] Rick Stout. The World Wide Web Complete Reference. Osborne McGraw-Hill, 1996.
- [12] The egcs project. EGCS GNU C Compiler. (URL <http://egcs.cygnum.com/>).
- [13] Thomas Boutell. CGI Programming in C & Perl. Addison-Wesley, 1996.
- [14] ZDNet. WebBench. (URL <http://www.zdnet.com/zdbop/webbench/webbench.html>).

จุฬาลงกรณ์มหาวิทยาลัย



ภาคผนวก

จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก.

โปรแกรมค้นหา SWISH-E ^[6]

SWISH-E มาจาก Simple Web Indexing System for Humans – Enhanced เป็นโปรแกรมค้นหาที่ใช้บนระบบปฏิบัติการ Unix ถูกสร้างขึ้นครั้งแรกโดย Kevin Hughes ใช้ชื่อว่า SWISH สำหรับใช้จัดเรียงและสืบค้นเอกสารสำหรับเพิ่มข้อความหรือเอกสารเฮกซ์ที่เอ็มแอล เนื่องจากในขณะนั้นโปรแกรมค้นหาที่มีอยู่มีความซับซ้อนมากและใช้งานยาก โปรแกรมค้นหา SWISH มีการใช้และเผยแพร่กันอย่างกว้างขวาง มีการปรับปรุงกันอย่างต่อเนื่อง จนกระทั่งประมาณปี ค.ศ. 1996 ห้องวิจัยของมหาวิทยาลัย Berkeley ได้รับอนุญาตจาก Kevin Hughes ให้สามารถปรับปรุงและแก้ไขส่วนที่เป็นปัญหาของโปรแกรมได้ โดยใช้ชื่อใหม่ว่า SWISH-E หลังจากนั้นทางมหาวิทยาลัย Berkeley ก็ได้มีการปรับปรุงโปรแกรมอย่างต่อเนื่อง และมีการเผยแพร่ต้นฉบับโปรแกรมอย่างไม่คิดมูลค่า มีการแลกเปลี่ยนความคิดเห็นกันระหว่างผู้ใช้กับทีมผู้พัฒนาระบบ

คุณสมบัติเบื้องต้น

- สร้างแฟ้มดัชนี สำหรับเพิ่มเอกสารเฮกซ์ที่เอ็มแอลและเพิ่มข้อความ
- ค้นหาข้อความ ทั้งแบบปกติ และแบบมีเงื่อนไข
 - สามารถใช้เครื่องหมายตรรก คือ AND , OR หรือ NOT มาใช้ร่วมกับข้อความที่ต้องการค้นหา
 - กรณีเอกสารเฮกซ์ที่เอ็มแอล สามารถเลือกที่จะค้นหาเฉพาะที่ได้ เช่น ค้นเฉพาะ head , title หรือ body ก็ได้ หรือ ค้นหาจาก META TAG ก็ได้

รูปแบบการใช้งาน

สำหรับสร้างแฟ้มดัชนี

```
Swish -i dir file -c file -f file -l file [-v (num)]
```

- i ชื่อแฟ้มหรือสถานที่เก็บ แฟ้มหรือกลุ่มแฟ้มที่ต้องการให้สร้างดัชนี
- c ชื่อโครงแบบ (Configuration) ของ SWISH-E
- f ชื่อแฟ้มดัชนีที่ต้องการสร้าง
- l ชื่อแฟ้มที่ต้องการบันทึกสถานะการทำงาน
- v ขอบเขตการแสดงผลสถานะการทำงาน (0-3) 0-ไม่แสดง 3-แสดงทุกขั้นตอน ถ้าไม่ระบุ โปรแกรมจะกำหนดค่าโดยปริยายคือ 3

สำหรับค้นหา

```
Swish -w word1 word2 -f file1 file2 -C file
```

- w ข้อความที่ต้องการค้นหา
- f แฟ้มดัชนี
- c ชื่อโครงสร้างของ SWISH-E

รูปแบบการค้นหา:

| | |
|-----------|---|
| W | ค้นหาเอกสารที่มีคำ w อยู่ |
| W1 w2 | ค้นหาเอกสารที่มีคำ w1 และ w2 อยู่ในเอกสารฉบับเดียวกัน |
| "w1 w2" | "" |
| W1+w2 | "" |
| w1 and w2 | "" |
| w1 or w2 | ค้นหาเอกสารที่มีคำ w1 หรือ w2 หรือ ทั้ง w1 และ w2 |
| w1 not w2 | ค้นหาเอกสารที่มีคำ w1 แต่ไม่มีคำ w2 |
| W* | ค้นหาเอกสารที่ขึ้นด้วยคำ w |

หมายเหตุ คำบางคำที่พบอยู่ในเอกสารหลายฉบับมากจนเกินไปเรียกว่า คำสามัญ (common word) จะถูกกำจัดออกจากแฟ้มดัชนีโดยอัตโนมัติ ดังนั้นคำจะพวกนี้ SWISH-E จะไม่ค้นหา

สำหรับรวมแฟ้มดัชนี

```
Swish -M index1 index2 ... output
```

- M รายชื่อแฟ้มดัชนี
- output แฟ้มดัชนีใหม่

สำหรับแสดงข้อมูลที่เก็บอยู่ในแฟ้มดัชนี

```
Swish -D file
```

- D ชื่อแฟ้มดัชนี

ทุกคำสั่งจะมีการแสดงผลทางอุปกรณ์แสดงผลมาตรฐาน ดังนั้นสามารถที่ทำการเปลี่ยนทิศทางไปยังปลายทางอื่นได้

ส่วนประกอบของโปรแกรม SWISH-E

SWISH-E เขียนขึ้นด้วยภาษาซี ประกอบด้วยแฟ้มต่างๆ ดังนี้

| แฟ้มส่วนหัว (Header file) | |
|---------------------------|--|
| ชื่อแฟ้ม | รายละเอียด |
| Check.h | ฟังก์ชันต้นแบบ (Prototype function) ของ แฟ้ม Check.c |
| Config.h | ตัวประมวลผลก่อน (Preprocessor) สำหรับค่าเริ่มต้นแฟ้มโครงสร้าง |
| Error.h | ฟังก์ชันต้นแบบสำหรับแฟ้ม Error.c |
| File.h | ฟังก์ชันต้นแบบสำหรับแฟ้ม File.c |
| Hash.h | ฟังก์ชันต้นแบบสำหรับแฟ้ม Hash.c |
| Index.h | ฟังก์ชันต้นแบบสำหรับแฟ้ม Index.c |
| List.h | ฟังก์ชันต้นแบบสำหรับแฟ้ม List.c |
| Mem.h | ฟังก์ชันต้นแบบสำหรับแฟ้ม Mem.c |
| Merge.h | ฟังก์ชันต้นแบบและแบบข้อมูลสำหรับแฟ้ม Merge.c |
| Search.h | ฟังก์ชันต้นแบบสำหรับแฟ้ม Search.c |
| String.h | ฟังก์ชันต้นแบบสำหรับแฟ้ม String.c |
| Swish.h | แฟ้มส่วนหัวหลักสำหรับ SWISH-E <ul style="list-style-type: none"> - กำหนดค่าเริ่มต้นตัวแปรหลักของระบบ - กำหนดชุดตารางตัวอักษร , Stop word |
| แฟ้มโปรแกรม | |
| ชื่อแฟ้ม | รายละเอียด |
| Check.c | ชุดฟังก์ชันตรวจสอบตัวอักษรและรหัสแฮชที่เอ็มแอล |
| Error.c | ฟังก์ชันแสดงค่าผิดพลาดและจบการทำงาน |
| File.c | ชุดฟังก์ชันตรวจสอบและจัดการเกี่ยวกับแฟ้ม |
| Hash.c | ชุดฟังก์ชันจัดการตารางค่าที่จัดทำดัชนี |
| Index.c | ชุดฟังก์ชันสำหรับสร้างแฟ้มดัชนี |
| List.c | ฟังก์ชันสำหรับสร้างรายการโยง (linked list) |
| Mem.c | ชุดฟังก์ชันจัดสรรและลบตัวแปรในหน่วยความจำ |
| Merge.c | ชุดฟังก์ชันสำหรับรวมแฟ้มดัชนี |
| Search.c | ชุดฟังก์ชันสำหรับค้นหาตามข้อความ |
| String.c | ชุดฟังก์ชันสำหรับตรวจสอบและจัดการเกี่ยวกับตัวแปรสายอักขระ |
| Swish.c | ฟังก์ชันหลักสำหรับกำหนดค่าเริ่มต้นและตรวจสอบอาร์กิวเมนต์ |

การปรับปรุงโปรแกรม SWISH-E ให้สามารถรองรับการประมวลผลตัวอักษรภาษาไทย

โปรแกรม SWISH-E^[10] ที่ปรับปรุงเป็นรุ่น 1.1 ใช้โปรแกรมตัวแปลภาษาซี ของ EGCS^[11] เนื่องจาก SWISH-E รุ่นที่นำมาปรับปรุงนี้ ถูกสร้างขึ้นเพื่อให้ใช้งานกับระบบปฏิบัติการ Unix จึงต้องมีการปรับปรุงให้ใช้งานกับระบบปฏิบัติการวินโดวส์แบบ 32 บิต และสามารถรองรับการประมวลผลตัวอักษรภาษาไทยได้ โดย

- เพิ่มชุดตัวอักษรภาษาไทยโดยเพิ่มกลุ่มตัวอักษรตั้งแต่ 'ก' เป็นต้นไป (เพิ่ม swish.h และ swish.c)
 - แถวลำดับ (Array) ของตัวอักษรที่สามารถทำดัชนีได้ (ตัวแปร indexchars)
 - แถวลำดับของตัวอักษรสำหรับสร้าง Map Table (ตัวแปร wordchars)
 - แถวลำดับของตัวอักษรสำหรับตรวจสอบตัวอักษรนำของคำที่จะทำดัชนี (ตัวแปร beginchars)
 - แถวลำดับของตัวอักษรสำหรับตรวจสอบตัวอักษรหลังสุดของคำที่จะทำดัชนี (ตัวแปร endchars)
- ปรับปรุงเงื่อนไขการเปรียบเทียบตัวอักษร โดยเพิ่ม cast (unsigned char) เพื่อเปรียบเทียบตัวอักษรโดยไม่คิดเครื่องหมาย (เพิ่ม string.c) เช่น


```
เดิม          if( ch == wordchar[l] )
      เปลี่ยนเป็น if( (unsigned char) ch == (unsigned char) wordchar[l] )
```
- ปรับปรุงเรื่องการเปิดปิดแฟ้ม โดยระบุให้เป็นการเปิดปิดแฟ้มแบบทวิภาค (Binary) และการอ่านข้อมูลจากแฟ้มแบบทีละตัวอักษรมาเป็นแบบอ่านทีละกลุ่ม (เพิ่ม index.c, merge.c และ search.c)

```

DEFINE_CLASSES
  CLASS: nn
  ...
DEFINE_REQUESTS
  CLASS:
    GET /file1
  ...

```

DEFINE_CLASSES กำหนดคลาสสำหรับทดสอบ
 CLASS: ชื่อคลาส
 nn เปอร์เซ็นต์ของคลาสที่จะใช้จำนวนเพื่อส่งคำร้องขอ
 DEFINE_REQUESTS กำหนดคำร้องขอ
 GET /file1 คำร้องขอ

ตัวอย่าง

```

DEFINE_CLASSES
  CLASS_A: 50
  CLASS_B: 30
  CLASS_C: 20
DEFINE_REQUESTS
  CLASS_A:
    GET /file1
  CLASS_B:
    GET /file2
  CLASS_C:
    GET /file3

```

ถ้ากำหนดให้ส่ง 100 คำร้อง โปรแกรม WebClient จะทำการส่ง :

1. GET /file1 จำนวน 50 คำร้อง
2. GET /file2 จำนวน 30 คำร้อง
3. GET /file3 จำนวน 20 คำร้อง

การกำหนดข้อมูลในแฟ้ม Test Suite

ข้อมูลในแฟ้ม Test Suite จะถูกบันทึกอยู่ในรูปแบบแฟ้มข้อความ ประกอบด้วย ชุดทดสอบที่เรียกว่า mix ซึ่งเป็นกลุ่มคำร้องขอและพารามิเตอร์ได้แก่

- MixName ชื่อ mix
- NumClients จำนวนเครื่องรับบริการใน mix
- RampUp กำหนดเวลาเริ่มส่งคำขอจนกระทั่งเริ่มทดสอบ (หน่วย: วินาที) โปรแกรมจะไม่นับที่กผลระหว่างช่วงนี้
- RampDown กำหนดเวลาจบการทดสอบจนกระทั่งเลิกส่งคำขอ (หน่วย: วินาที) โปรแกรมจะไม่นับที่กผลระหว่างช่วงนี้
- Length กำหนดเวลาที่ใช้ทดสอบในแต่ละ mix ซึ่งต้องมากกว่า RampUp และ RampDown รวมกัน (หน่วย: วินาที)
- Delay เวลาที่ให้ตัวรับบริการเว็บรอก่อนเริ่มทดสอบ (หน่วย: วินาที)
- Think เวลาที่ตัวรับบริการเว็บรอก่อนที่จะส่งคำร้องถัดไปหลังจากได้รับการตอบสนอง (หน่วย: วินาที)
- Number of Threads จำนวนสายใยที่ต้องการให้ตัวรับบริการเว็บใช้ระหว่างทดสอบ
- Thread Inactivity Timeout กำหนดเวลาสูงสุดที่จะรอการตอบสนองจากเครื่องบริการ
- Receive Buffer Size ขนาดข้อมูลสูงสุดที่ตัวรับบริการเว็บสามารถอ่านจากตัวบริการเว็บใน 1 ครั้ง (หน่วย: ไบต์)
- WWW Server ชื่อเครื่องของตัวบริการเว็บ
- WWW Port ช่องทางของเว็ลด์ไวด์เว็บปกติจะเท่ากับ 80
- WorkLoad File ระบุชื่อแฟ้ม workload

ภาคผนวก ค.

ตัวอย่างรายงานบทคัดย่อ ฯ และการแสดงผล

TOPIC:: Treatment of activated sludge process effluent for cooling tower make-up water / Somnuk

Jarudilokkul

IDNO:: B2820638

TNAME:: สมนึก จารุติลกกุล

ENAME:: Somnuk Jarudilokkul

ADVISOR:: Wiwut Tanthapanichakoon

DEPARTMENT:: Thesis (M.Eng.)—Chulalongkorn University, 1995

YEAR:: 1995

KEYWORD:: Sewage|Purification|Activated sludge process

THAI ABSTRACT:: การบำบัดน้ำที่ออกจากกระบวนการสลัดจ์กัมมันต์ เพื่อให้ได้คุณภาพของน้ำเติมหอทำน้ำเย็น จะทำให้สามารถใช้ประโยชน์เป็นแหล่งน้ำได้ การศึกษานี้เกี่ยวกับการพัฒนาวิธีการบำบัดขั้นสูงของน้ำที่ออกจากกระบวนการสลัดจ์กัมมันต์ ระบบบำบัดประกอบด้วย การกรองด้วยวัสดุเม็ด 2 ชนิด การแลกเปลี่ยนไอออน และเบรคพ้อยต์คลอรีเนชัน จากผลการวิจัยพบว่า เยื่อไนท์ที่เหมาะสมสำหรับหน่วยปฏิบัติการแต่ละหน่วยเป็นดังนี้ เยื่อไนท์การกรองด้วยวัสดุเม็ด 2 ชนิด ได้แก่ ชั้นทรายซึ่งมีความสูง 20 ซม. และชั้นแอนทราไซต์ซึ่งมีความสูง 40 ซม. และอัตราการไหลของน้ำเข้า 20 ลบ.ม./ตร.ม.-ชม. เยื่อไนท์การแลกเปลี่ยนไอออน ได้แก่ ชั้นเรซิน Duolite C20 ซึ่งมีความสูง 90 ซม. และอัตราการไหลของน้ำเข้า 10 ลบ.ม./ตร.ม.-ชม. เยื่อไนท์ของเบรคพ้อยต์คลอรีเนชัน ได้แก่ ปริมาณโซเดียมไฮโปคลอไรต์ที่เติมตอนแรก 2.7-3.0 มก.-ล. ลำดับของปฏิบัติการในระบบเป็นดังนี้ เริ่มต้นจากการกรองด้วยวัสดุเม็ดเพื่อกำจัดของแข็งแขวนลอย จากนั้นผ่านน้ำเข้าสู่การแลกเปลี่ยนไอออนเพื่อกำจัดความเป็นด่าง น้ำที่ออกมาจะมีก๊าซละลายอยู่จึงนำไปผ่านหอไล่ก๊าซเพื่อกำจัดก๊าซ น้ำจากหอไล่ก๊าซมีสภาพเป็นกรดจึงนำมาสะเทินด้วยน้ำจากการกรองด้วยวัสดุเม็ดในสัดส่วนที่เท่ากัน สุดท้ายนำน้ำที่ผสมแล้วมาเติมโซเดียมไฮโปคลอไรต์เพื่อกำจัดเชื้อจุลินทรีย์ทั้งหมดและเชื้อคอลลีฟอร์ม ผลการวิเคราะห์คุณภาพน้ำพบว่า น้ำที่ออกจากระบบดังกล่าวมีคุณภาพเหมาะสมต่อการเติมในหอทำน้ำเย็น

ENG ABSTRACT:: Further treatment of activated sludge effluent to meet make-up water quality requirements for cooling tower can render it usable as water supply. This study pertains to the development of advanced treatment of activated sludge effluent. The process consists of dual-media filtration, ion-exchange and break-point chlorination. It has been found that suitable conditions of each unit are as follows. Dual-media filtration conditions are sand height 20 cm., anthracite height 40 cm. and influent flow rate 20 m³/m²-hr. Ion-exchange conditions are Duolite C20 resin height 90 cm. and influent flow rate 10 m³/m²-hr. Break-point chlorination condition is initial sodium hypochlorite dosage 2.7-3.0 mg/l. The sequence of operation is as follows. First, suspended solids were removed by the dual-media filter. Second, alkalinity was removed by ion-exchanger. Third, dissolved gas in the ion-exchange effluent was removed by the degasifier. Fourth, pH of the degasifier effluent was neutralized by mixing with the dual-media effluent at a ratio 1:1. Finally, total plate count and coliforms were killed by break-point chlorination. The quality of the effluent water from process meets the specifications of cooling tower make-up water.

~~~~~



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

การแสดงผลในรูปแบบของเว็บเพจ

|                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |            |      |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|------|
| ชื่อเรื่อง         | Treatment of activated sludge process effluent for cooling tower make-up water / สมบูรณ์ จารุดีล็อกกุล                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |            |      |
| จัดทำโดย           | สม. นี. จารุดีล็อกกุล<br>Somnuk Jarudilokkul                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |            |      |
| เลขทะเบียนคำ       | B2820038                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | ปีที่จัดทำ | 1995 |
| สาขาที่ปรึกษา      | Water Treatment/Thonon                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |            |      |
| คำสำคัญ            | Sewage/Purification/activated sludge process                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |            |      |
| บทคัดย่อภาษาไทย    | <p>การบำบัดน้ำทิ้งออกจากขบวนการบำบัดที่มีผลทำให้คุณภาพของน้ำดีพอที่จะนำมาใช้ประโยชน์เป็นแหล่งน้ำได้ การบำบัดน้ำทิ้งจากคอกหมูของมหาวิทยาลัยเกษตรศาสตร์วิทยาเขตลำปางมีพื้นที่ ๒ ไร่ ๒ งาน และขบวนการบำบัดน้ำทิ้งมี 2 ขั้นตอนแรกคือกรองโคลน และตกตะกอนด้วยวิธีตกตะกอนแบบลอยน้ำ เป็นขั้นที่เฉพาะสำหรับหน่วยปฏิบัติการแต่ละแห่งเป็นต้นไปคือได้แก่ การกรองด้วยวิธีตกตะกอน 2 ขั้นตอน ขั้นแรกที่มีผิวสูง 20 ซม. และชั้นแฉะทรายที่มีผิวสูง 40 ซม. และอัตราการไหลของน้ำที่ 20 ลบ.ม./คร.ม.-ชม. ส่วนขั้นตอนที่สองคือใช้เรซิน Duolite C20 ซึ่งมีความสูง 90 ซม. และอัตราการไหลของน้ำที่ 10 ลบ.ม./คร.ม.-ชม. เรซินซึ่งสามารถพักกรองน้ำทิ้งได้ประมาณ ๒-3 วัน โดยปกติได้มีผลกรอง 7-30 มก./ลิ. ขั้นตอนปฏิบัติการในระยะต่อไปได้มีเริ่มต้นจากการกรองด้วยวิธีตกตะกอนแบบลอยน้ำตามด้วยเครื่องแลกเปลี่ยนไอออนเพื่อขจัดความเค็มต่าง มีที่ออกสารนี้ก็จะลงอยู่ถึงน้ำในขั้นตอนต่อไปซึ่งมีเครื่องกำจัดก๊าซ ฟองอากาศใช้ก๊าซมีเทนเป็นชนิดจึงนำมาใช้เป็นตัวกรองการกำจัดออกซิเจน คในถังสามเหลี่ยมสูงที่ใช้น้ำที่ผ่านการบำบัดแล้ว ไอเดียมีใช้ไปของไหลที่กำจัดเชื้อจุลินทรีย์ที่ยังมีและเชื้อจุลินทรีย์มีผลกำจัดน้ำทิ้ง น้ำที่ออกจากระบบดังกล่าวมีคุณภาพเหมาะสมต่อการนำไปใช้งานได้</p>                                                                                                                                                                               |            |      |
| บทคัดย่อภาษาอังกฤษ | <p>Further treatment of activated sludge effluent to meet make-up water quality requirements for cooling tower can render it usable as water supply. This study pertains to the development of advanced treatment of activated sludge effluent. The process consists of dual-media filtration, ion-exchange and break-point chlorination. It has been found that suitable conditions of each unit are as follows: Dual-media filtration conditions are sand height 20 cm., anthracite height 40 cm. and influent flow rate 20 m<sup>3</sup>/m<sup>2</sup>-hr. Ion-exchange conditions are Duolite C20 resin height 90 cm. and influent flow rate 10 m<sup>3</sup>/m<sup>2</sup>-hr. Break-point chlorination conditions are initial sodium hypochlorite dosage 27-30 mg/l. The sequence of operations is as follows: First, suspended solids were removed by the dual-media filter. Second, alkalinity was removed by ion-exchanger. Third, dissolved gas in the ion-exchange effluent was removed by the degasifier. Fourth, pH of the degasifier effluent was neutralized by mixing with the dual-media effluent at a ratio 1:1. Finally, total plate count and coliforms were killed by break-point chlorination. The quality of the effluent water from process meets the specifications of cooling tower make-up water.</p> |            |      |

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ง.

แสดงข้อความที่ใช้ทดสอบการค้นคืนตามบทที่ 6

การค้นคืนข้อความ "ดิลก"

ระบบสนับสนุนการค้นคืนบทคัดย่อวิทยานิพนธ์ผ่านเครือข่าย  
Thesis Retrieval Supported System Via Network

ระบุข้อความที่ใช้ทดสอบการค้นหา

ค้นหา  ค้นหา

ค้นหาเฉพาะ  หัวข้อ  คำสำคัญ  บทคัดย่อ  คำนำ

แสดงจำนวน: 10 รายการ

ผลการค้นหา "ดิลก" พบ 2 รายการ

|   |                          |                                                                                                                           |
|---|--------------------------|---------------------------------------------------------------------------------------------------------------------------|
| 1 | <a href="#">s0001055</a> | Treatment of activated sludge process effluent for cooling tower make-up water / Somrak Jarulloklut                       |
| 2 | <a href="#">s0000001</a> | An analysis of the development of characteristics of citizenship in Thai compulsory education history / Uksavas Preedolok |

การค้นคืนข้อความ "ศึกษา"

ระบบสนับสนุนการค้นคืนบทคัดย่อวิทยานิพนธ์ผ่านเครือข่าย  
Thesis Retrieval Supported System Via Network

ระบุข้อความที่ใช้ทดสอบการค้นหา

ค้นหา  ค้นหา

ค้นหาเฉพาะ  หัวข้อ  คำสำคัญ  บทคัดย่อ  คำนำ

แสดงจำนวน: 10 รายการ

ผลการค้นหา "ศึกษา" พบ 1172 รายการ

1 12 3 4 5 8 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 ] (ต่อไป)

|    |                          |                                                                                                                                                                                                                                                 |
|----|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | <a href="#">s0000001</a> | An analysis of the development of characteristics of citizenship in Thai compulsory education history / Uksavas Preedolok                                                                                                                       |
| 2  | <a href="#">s0000003</a> | Analysis of problems that students face in using the Internet to access Learning Management System / Chan, Anonwat                                                                                                                              |
| 3  | <a href="#">s0000004</a> | Opinions of administration and teachers concerning the management of school public relations activities in large secondary schools under the Jurisdiction of the Department of General Education, Educational Region Seven / Chidchar Chirapred |
| 4  | <a href="#">s0000005</a> | Some of problems of the students in the learning process in the case of Burangkajit Metropolitan / Wiratham Kitiraporn                                                                                                                          |
| 5  | <a href="#">s0000006</a> | State and problems of the operation concerning the maintenance of private higher education institution instructors in regional area / Somsoorn Sampradit                                                                                        |
| 6  | <a href="#">s0000007</a> | การศึกษารูปแบบการดำเนินงานของศูนย์ส่งเสริมและพัฒนาคุณภาพชีวิตคนพิการแห่งชาติเขตที่ 10 / สุชาติชาย งามกุล                                                                                                                                        |
| 7  | <a href="#">s0000008</a> | A study of the operation of educational innovation and technology development of educational supervisors under the jurisdiction of the Office of the National Primary Education Commission / Anurawn Ganout                                     |
| 8  | <a href="#">s0000009</a> | การศึกษารูปแบบการดำเนินงานของศูนย์ส่งเสริมและพัฒนาคุณภาพชีวิตคนพิการแห่งชาติเขตที่ 10 / สุชาติชาย งามกุล                                                                                                                                        |
| 9  | <a href="#">s0000010</a> | Food consumption behaviors of Prathom Suksa 56 students under the jurisdiction of the Office of the National Primary Education Commission in the Northeastern Region / Witsana Fragobvong                                                       |
| 10 | <a href="#">s0000011</a> | The problem of learning difficulties in mathematics of the students in the primary school / Uksavas Preedolok                                                                                                                                   |



การค้นคืนข้อความ "คอมพิวเตอร์+PC\* not UNIX"

**ระบบสืบค้นการค้นคืนบทความวิทยานิพนธ์ผ่านเครือข่าย**  
*Thesis Retrieval Supported System Via Network*

รายชื่อเอกสารที่สืบค้นได้

คอมพิวเตอร์+PC\* not UNIX    ค้นหา    ลบ

จำนวนหน้า 10 (รายการ)

**ผลการค้นหา 'คอมพิวเตอร์+PC\* not UNIX' พบ 19 รายการ**

[ 12 ] (ต่อไป)

|    |                          |                                                                                                                                            |
|----|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | <a href="#">๑0001532</a> | Bioequivalence of ceftriaxone intramuscular injections / Orawan Srisulachai                                                                |
| 2  | <a href="#">๑0000695</a> | A study of current status and trend of the Buddhist Based System (BBS) Thailand / Phrasachai Wachirapattanasri                             |
| 3  | <a href="#">๑0001327</a> | Design and development of an IBM PC based microcomputer training set / Taweechai Charoensudthan                                            |
| 4  | <a href="#">๑0001595</a> | A study on factors of homicidal psychosis: patients related to court sentences / Phitsyaret Saksitanamichai                                |
| 5  | <a href="#">๑0000488</a> | Mental health of public service motorcycle drivers in Bangkok Metropolis / Phachangrit Phapoom                                             |
| 6  | <a href="#">๑0001508</a> | Bangkok Buddhist students' media exposure, expectation and gratification from Buddhism talk show on television / Sirinimee Pongpraditsakul |
| 7  | <a href="#">๑0001515</a> | Interactive geographic information system for tourism (a case study of Changwat Kanchanaburi) /                                            |
| 8  | <a href="#">๑0001125</a> | Factor influencing gratification of cable television / Kribsraek Pacharattana                                                              |
| 9  | <a href="#">๑0001198</a> | Attitudes towards uses and gratifications of 'sornthra seesathanee panga banniang' of television watchers / Pattanavadee Lovchit           |
| 10 | <a href="#">๑0001613</a> | Factors affecting University students in listening to P.M. 104.5 Muz. Green Wave radio program / Pattana Chaitanjan                        |



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## ภาคผนวก จ.

แสดงตัวอย่างการค้นคืนแบบต่างๆ

ตัวอย่างข้อความถาม ในบทนี้มาจากข้อมูลในภาคผนวก ค.

การค้นคืนโดยกำหนดให้ค้นหาเฉพาะหัวข้อเรื่อง

ระบบสนับสนุนการค้นคืนบทคัดย่อวิทยานิพนธ์ผ่านเครือข่าย  
Thesis Retrieval Supported System Via Network  
ระบุชื่อมหาวิทยาลัยที่ต้องการ  
Treatment of activated ค้นหา ลบ

ในบทเฉพาะ  หัวข้อ  ชื่อผู้แต่ง  สาขาวิชาที่เกี่ยวข้อง  คำสำคัญ

แสดงเฉพาะที่ 10 รายการ

ผลการค้นหา 'Treatment of activated' พบ 1 รายการ

|   |          |                                                                                                     |
|---|----------|-----------------------------------------------------------------------------------------------------|
| 1 | 62820838 | Treatment of activated sludge process effluent for cooling tower make-up water / Somnuk Jaruchitkul |
|---|----------|-----------------------------------------------------------------------------------------------------|

การค้นคืนโดยกำหนดให้ค้นหาเฉพาะชื่อผู้แต่ง

ระบบสนับสนุนการค้นคืนบทคัดย่อวิทยานิพนธ์ผ่านเครือข่าย  
Thesis Retrieval Supported System Via Network  
ระบุชื่อมหาวิทยาลัยที่ต้องการ  
สมนึก จารุชิตกุล ค้นหา ลบ

ในบทเฉพาะ  หัวข้อ  ชื่อผู้แต่ง  สาขาวิชาที่เกี่ยวข้อง  คำสำคัญ

แสดงเฉพาะที่ 10 รายการ

ผลการค้นหา 'สมนึก จารุชิตกุล' พบ 1 รายการ

|   |          |                                                                                                     |
|---|----------|-----------------------------------------------------------------------------------------------------|
| 1 | 62820838 | Treatment of activated sludge process effluent for cooling tower make-up water / Somnuk Jaruchitkul |
|---|----------|-----------------------------------------------------------------------------------------------------|

จุฬาลงกรณ์มหาวิทยาลัย

การค้นคืนโดยกำหนดให้ค้นหาเฉพาะชื่ออาจารย์ที่ปรึกษา

ระบบสนับสนุนการค้นคืนบทความวิทยานิพนธ์ผ่านเครือข่าย  
Thesis Retrieval Supported System Via Network

ระบุชื่ออาจารย์ที่ปรึกษาลำดับที่  ค้นหา ค้นหา

จำนวนหน้า: 10

ผลการค้นหา 'Wiwut Tanthapanichakoon' พบ 3 รายการ

|   |          |                                                                                                                                        |
|---|----------|----------------------------------------------------------------------------------------------------------------------------------------|
| 1 | B2805303 | Development and construction of tester and testing of Lower Explosion Limit of particulate materials / Songchai Wriyaunpeewong         |
| 2 | B2820638 | Treatment of activated sludge process effluent for cooling tower make-up water / Jannat Anurupattana                                   |
| 3 | B2812083 | Factors influencing the dispersion of organic pigments in polyethylene upon using a continuous twin-screw kneader / Aranya Thongchasee |

การค้นคืนโดยกำหนดให้ค้นหาเฉพาะคำสำคัญ

ระบบสนับสนุนการค้นคืนบทความวิทยานิพนธ์ผ่านเครือข่าย  
Thesis Retrieval Supported System Via Network

ระบุชื่ออาจารย์ที่ปรึกษาลำดับที่  ค้นหา ค้นหา

จำนวนหน้า: 10

ผลการค้นหา 'Sewage' พบ 6 รายการ

|   |          |                                                                                                                               |
|---|----------|-------------------------------------------------------------------------------------------------------------------------------|
| 1 | B2801449 | B2801449.html                                                                                                                 |
| 2 | B2824528 | Heavy metal recovery by sulfidation and chromatography using ethanoglycerol based PVA water hydrogel / Kuntorn Akhavanisri    |
| 3 | B2805101 | Solidification of heavy metal sludge from COG waste water treatment by using cement and lignite fly ash / Narunil Kriwan      |
| 4 | B2820638 | Treatment of activated sludge process effluent for cooling tower make-up water / Jannat Anurupattana                          |
| 5 | B2839394 | Removal of organics and colour of obnoxious slip by anaerobic and electrochemical treatment process / from Dusadeepun         |
| 6 | B338212  | Effect of C60 fullerenes on nitrogen removal from domestic wastewater by using activated sludge process / Jannat Anurupattana |

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

การค้นคืนโดยระบุข้อความร่วมกับ "\*" และกำหนดให้ค้นหาเฉพาะชื่อผู้แต่งหรืออาจารย์ที่ปรึกษา

**ระบบสืบค้นการค้นคืนบัณฑิตยศึกษานิพนธ์ผ่านเครือข่าย**  
*Thesis Retrieval Supported System Via Network*

ระบุชื่อค้นคืนที่ ต้องการสืบค้น

ค้นหา  ฟิลด์  รายการสืบค้น  ฟิลด์อื่น

เลขหมายสืบค้น

ผลการค้นหา 'พว\*' พบ 13 รายการ

[ 1 2 ] [ ต่อหน้า ]

|    |          |                                                                                                                                                                                         |
|----|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | B2796466 | Evaluation program of prevention and control of dengue hemorrhagic fever in primary school, Amphoe Mueang, Nakhon Ratchasima province / Nuantra-oi Wuestornphan                         |
| 2  | B281383x | Effect of granular activated carbon on the adsorption of organic pollutants in water / Chindit Sornvith / Wimon 13 2552:66-67                                                           |
| 3  | B2766648 | Characteristics of elementary school teachers under the jurisdiction of the Bangkok Metropolitan administration in globalization era as perceived by themselves / Sudasinee Witsakanond |
| 4  | B2817693 | LaTeX - ระบบสืบค้นที่รองรับการสืบค้นแบบเต็มรูปแบบ / Linda Pichayak                                                                                                                      |
| 5  | B2838886 | Effect of metal salt on physical properties of PVA film / Suphanee Fongthitwitoon                                                                                                       |
| 6  | B2805303 | Development and characterization of thin membranes of low(er) expansion limit of polyamide materials / Songchai Wityaoungsakong                                                         |
| 7  | B2820638 | Treatment of activated sludge process effluent for cooling tower make-up water / Somrak Jarudilokul                                                                                     |
| 8  | B2797422 | Development of web-based system for air quality software / Kerasak Rangsiwanrat                                                                                                         |
| 9  | B2812083 | Factors influencing the dispersion of organic pigments in polyethylene upon using a continuous twin-screw kneader / Aranya Thongchiew                                                   |
| 10 | B2837468 | Development of a control system for feed composition / Malinee Manpracet                                                                                                                |

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## ภาคผนวก จ.

### รายละเอียดแฟ้มรหัสต้นฉบับ

ระบบสนับสนุนการค้นคืนบทคัดย่อวิทยานิพนธ์ผ่านเครือข่าย

ระบบสนับสนุนการค้นคืนบทคัดย่อวิทยานิพนธ์ผ่านเครือข่ายประกอบด้วย 3 ส่วนงานหลักดังนี้

#### 1. ส่วนจัดเตรียมข้อมูล ประกอบด้วย

##### 1.1 ฟังก์ชันสำหรับจัดทำดัชนีคำ ประกอบด้วยแฟ้มต่างๆ ดังนี้

- |        |             |                                            |
|--------|-------------|--------------------------------------------|
| 1.1.1  | global.h    | กำหนดค่าเริ่มต้นฟังก์ชันจัดทำดัชนีคำ       |
| 1.1.5  | routine.h   | กำหนดฟังก์ชันต้นแบบของฟังก์ชันจัดทำดัชนีคำ |
| 1.1.6  | tctype.h    | กำหนดฟังก์ชันต้นแบบของฟังก์ชันจัดทำดัชนีคำ |
| 1.1.7  | thai.h      | กำหนดรหัสตัวอักษรภาษาไทย                   |
| 1.1.2  | thaiword.h  | กำหนดฟังก์ชันต้นแบบของแฟ้มจัดทำดัชนีคำ     |
| 1.1.8  | thaiutil1.c | ฟังก์ชันตรวจสอบการจัดทำดัชนีคำแฟ้มที่ 1    |
| 1.1.9  | thaiutil2.c | ฟังก์ชันตรวจสอบการจัดทำดัชนีคำแฟ้มที่ 2    |
| 1.1.10 | thaiutil3.c | ฟังก์ชันตรวจสอบการจัดทำดัชนีคำแฟ้มที่ 3    |
| 1.1.11 | thaiutil4.c | ฟังก์ชันตรวจสอบการจัดทำดัชนีคำแฟ้มที่ 4    |
| 1.1.12 | thaiutil5.c | ฟังก์ชันตรวจสอบการจัดทำดัชนีคำแฟ้มที่ 5    |

แฟ้มเหล่านี้ไม่ได้ถูกปรับปรุงเพิ่มเติมในงานวิจัยนี้ ดังนั้นผู้ที่สนใจสามารถค้นคว้าได้จากงานวิจัย ระบบการค้นคืนข้อความภาษาไทยโดยใช้แฟ้มข้อมูลผกผัน<sup>[3]</sup>

##### 1.2 โปรแกรมจัดเตรียมแฟ้มข้อมูล (gnmain.exe) ประกอบด้วยแฟ้มต่างๆ ดังนี้

- |        |            |                                                     |
|--------|------------|-----------------------------------------------------|
| 1.2.1  | gnconfig.h | กำหนดค่าเริ่มต้นฟังก์ชันหลัก                        |
| 1.2.2  | gnmem.h    | กำหนดฟังก์ชันต้นแบบของแฟ้ม gnmem.c                  |
| 1.2.3  | gnstring.h | กำหนดฟังก์ชันต้นแบบของแฟ้ม gnstring.h               |
| 1.2.4  | gnfile.c   | จัดการเกี่ยวกับการอ่าน/เขียนข้อมูล                  |
| 1.2.5  | gnmain.c   | ระบบงานหลัก                                         |
| 1.2.6  | gnmem.c    | จัดการเกี่ยวกับหน่วยความจำ                          |
| 1.2.7  | gnread.c   | จัดการเกี่ยวกับการอ่านข้อมูลจากแฟ้ม                 |
| 1.2.8  | gnstring.c | จัดการเกี่ยวกับค่าสายอักขระ                         |
| 1.2.9  | thaiword.c | ฟังก์ชันจัดทำดัชนีคำ ที่ได้รับการปรับปรุงให้เหมาะสม |
| 1.2.10 | makefile   | ข้อกำหนดเกี่ยวกับการแปลโปรแกรม (Compile)            |

1.3 โปรแกรมแบ่งคำ (tsepword.exe) จะเรียกใช้ฟังก์ชันจัดทำดัชนีคำ โดยข้อกำหนดผ่านตัวประมวลผลก่อนในแฟ้ม thaiword.c ซึ่งอยู่ในส่วนโปรแกรมจัดเตรียมแฟ้มข้อมูลโดยเปลี่ยนแปลงรายละเอียดของการแปลโปรแกรมดังนี้

1.3.1 tsepword.mak ข้อกำหนดเกี่ยวกับการแปลโปรแกรม

2. ส่วนติดต่อกับผู้ใช้ ประกอบด้วยแฟ้มต่างๆ ดังนี้

- 2.1 admin.htm เว็บเพจหลักสำหรับการจัดการระบบ
- 2.2 admin\_h.htm เว็บเพจส่วนหัวสำหรับ admin.htm
- 2.3 index.htm เว็บเพจแสดงหัวข้อเลือกเริ่มต้น
- 2.4 search.htm เว็บเพจหลักสำหรับการระบุข้อความเพื่อค้นคืน
- 2.5 s\_help.htm เว็บเพจแสดงรายละเอียดช่วยเหลือสำหรับข้อความเพื่อค้นคืน
- 2.6 s\_main.htm เว็บเพจสำหรับระบุข้อความเพื่อค้นคืน
- 2.7 login.htm เว็บเพจที่เรียกโดย admin.pl เพื่อระบุชื่อผู้ใช้และรหัสผ่าน
- 2.8 menu.htm เว็บเพจที่เรียกโดย admin.pl เพื่อแสดงหัวข้อการทำงาน

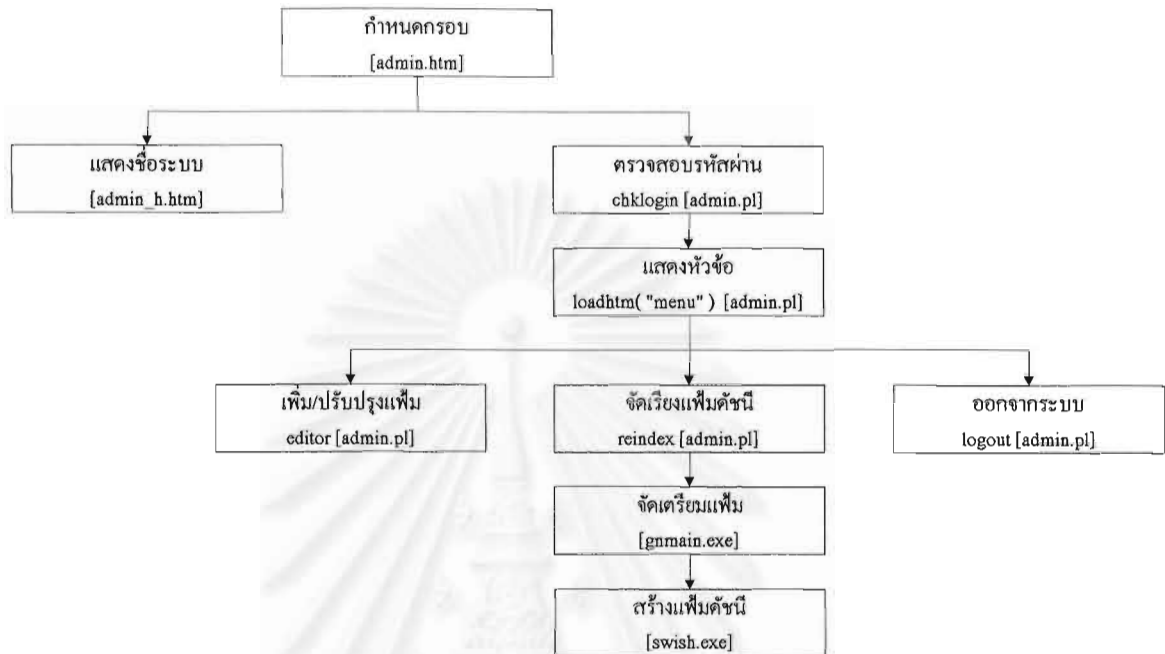
3. โปรแกรมบทคำสั่งซีไอ ประกอบด้วยแฟ้มต่างๆ ดังนี้

- 3.1 search.pl สำหรับประมวลผลข้อความสำหรับค้นคืน
- 3.2 admin.pl สำหรับประมวลผลการจัดเตรียมแฟ้มข้อมูลแบบระยะไกล
- 3.3 mylib.pl แฟ้มห้องสมุดสำหรับเรียกใช้โดย search.pl และ admin.pl

### รูปแสดงการเรียกใช้โปรแกรมย่อย

(หมายเหตุ ตัวอักษรที่อยู่ใน [ .. ] หมายถึงชื่อแฟ้มเก็บโปรแกรมย่อย)

ส่วนติดต่อกับผู้ใช้และการเรียกใช้โปรแกรมบทคำสั่งซีไอสำหรับจัดการระบบ



ส่วนติดต่อกับผู้ใช้และการเรียกใช้โปรแกรมบทคำสั่งซีไอสำหรับค้นคืน



## รายละเอียดเพิ่มเติมฉบับ

โปรแกรมจัดเตรียมเพิ่มข้อมูล

1.2.1 gnconfig.h กำหนดค่าเริ่มต้นฟังก์ชันหลัก

```
/* gnconfig.h
   Generated program By STK */
```

```
#ifndef _GNCONFIG_H
#define _GNCONFIG_H
```

```
#include <stdio.h>
#include <dos.h>
#include <string.h>
#include "gnstring.h"
```

```
char tolower(char);
```

```
#define MAX_STRING_LEN 4096
#define MAX_FILE_NAME 80
#define MAX_FIELD 50
#define MAX_FIELD_LEN 100
#define MAX_FLAG_LEN 20
#define MAX_DELIFIELD 10
#define BUFFER_SIZE 10240
```

```
#define TRUE 1
#define FALSE 0
```

```
struct String {
    char line[ MAX_STRING_LEN ];
    unsigned char id;
    struct String *next;
};
```

```
struct sortedString {
    char line[ MAX_STRING_LEN ];
    struct sortedString *right;
    struct sortedString *left;
};
```

```
struct fieldProperty {
    char name[ MAX_FIELD_LEN ];
    char flag[ MAX_FLAG_LEN ];
    int size;
    int start;
};
```

```
#define ID_STR_FIELD 1
#define ID_STR_TEMPLATE 2
#define ID_STR_BUFFER 3
```

```
#ifdef MAIN_FILE
    #define VAR
#else
    #define VAR extern
#endif
```



```

VAR char  fileOfField[ MAX_FILE_NAME ];
VAR struct fieldProperty field[ MAX_FIELD ];

VAR char  deliField[ MAX_DELIFIELD ];
VAR char  deliRecord[ MAX_FIELD_LEN ];

VAR int   totalField;

VAR char  fileOfTemplate[ MAX_FILE_NAME ];
VAR struct String *templateList;
VAR char  fileOfText[ MAX_FILE_NAME ];
VAR char  writeto[ MAX_STRING_LEN ];

VAR long  counter;
VAR long  initCounter;
VAR int   sizeOfCounter;

VAR int   uniqueValue;

VAR struct sortedString *fieldSort[ MAX_FIELD ];
VAR char  fileOfUnique[ MAX_FILE_NAME ];
VAR char  fileOfLog[ MAX_FILE_NAME ];
VAR FILE  *fpLog;

void procerr(char *); // gnmain.c
void readtext(); // gnread.c
void parseField( char *buff , char *name , char *flag , int *size , int *start ); // gnmain.c

#endif

```

### 1.2.2 gnmem.h กำหนดฟังก์ชันต้นแบบของแฟ้ม gnmem.c

```

/* gnmem.h */
void *emalloc();
void *erealloc();
char *memcpy(char *,char *,int,int,int );

```

### 1.2.3 gnstring.h กำหนดฟังก์ชันต้นแบบของแฟ้ม gnstring.h

```

/* gnstring.h */

struct String *addtoString( unsigned char , struct String *, char *);
char *strAllTrim( char *);
char *strLower( char *);
char *lstrstr(char *,char *);
char *nstrstr(char *,char *);
char *getword(char *,int *);
struct sortedString *newSort(char *);

```

### 1.2.4 gnfile.c จัดการเกี่ยวกับการอ่าน/เขียนข้อมูล

```

/* gnfile.c */
#include "gnconfig.h"

int freadln( FILE * , char * );
int freadword( FILE * , char * );
int freadstop( FILE * , char * , char );

int freadln( FILE *fp , char *buffer ) {
    return freadstop( fp , buffer , '\n' );
}

int freadword( FILE *fp , char *buffer ) {
    char ch;
    int pos = 0;
    while( 1 ) {
        ch = fgetc( fp );

        if( ch == ' ' && pos == 0 )
            continue;

        if( ch == '\r' )
            continue;

        if( ch == EOF || ch == '\n' || ch == ' ' || ch == '\t' ||
            (unsigned char) ch == 160 )
            break;

        buffer[pos++] = (unsigned char) ch;
    }

    if( ch == EOF && pos == 0 )
        return -1;

    return pos;
}

int freadstop( FILE *fp , char *buffer , char stopchar ) {
    char ch;
    int pos = 0;
    while( 1 ) {
        ch = fgetc( fp );
        if( ch == '\r' )
            continue;
        if( ch == EOF || (unsigned char) ch == (unsigned char) stopchar )
            break;
        buffer[pos++] = (unsigned char) ch;
    }

    buffer[pos] = '\0';
    if( ch == EOF && pos == 0 )
        return -1;
    return pos;
}

```

## 1.2.5 gnmain.c ระบบงานหลัก

```

/* gnmain.c

    Main program of gnmain.exe

*/

#define MAIN_FILE

#include "gnconfig.h"
#include <dir.h>

int main(int argc , char *argv[] ) {
int nPos;

    if( argc == 1 ) {
        usage();
    }

    /* --- initial default value --- */

    templateList = NULL;

    for( nPos = 0; nPos < MAX_FIELD; nPos++ ) {
        field[nPos].name[0] = field[nPos].flag[0] = '\0';
        field[nPos].size = 0; // default unlimited
        fieldSort[nPos] = NULL; // for list unique field & sort
    }

    fileOfField[0] = fileOfTemplate[0] = fileOfText[0] = fileOfLog[0] = '\0';
    strcpy( fileOfField , "field.lst"); // default
    strcpy( deliField , ":@" ); // default delimiter
    deliRecord[0] = '\0';
    fpLog = stdout; // file pointer of log file

    fileOfUnique[0] = writeto[0] = '\0';

    counter = initCounter = 0;
    sizeofCounter = 4; // default length

    uniqueValue = 0; // Use line unique field */
    totalField = 0; // Total field */

    /* --- End of initial default value --- */

    nPos = 1;
    while( 1 ) {
        if( nPos >= argc )
            break;

        if( argv[nPos][0] != '-' || argv[nPos][2] != '\0' ) {
            /* ignore invalid parameter */
            nPos++;
            continue;
        }
    }

```

```

if( nPos+1 >= argc )
    break;

switch( argv[nPos][1] ) {

    case 'f' : strcpy( fileOfField , argv[nPos+1] );
                break;

    case 'p' : strcpy( fileOfTemplate , argv[nPos+1] );
                break;

    case 't' : strcpy( fileOfText , argv[nPos+1] );
                break;

    case 'l' : strcpy( fileOfLog , argv[nPos+1] );
                break;

    case 'T' : initialize( argv[nPos+1] );
                break;

    default:

        printf("unknow parameter [%s]\n",argv[nPos] );
        exit(1);
}

nPos += 2;
}

if( fileOfLog[0] != '\0' ) {

    fpLog = (FILE *) fopen( fileOfLog , "a" );

    if( fpLog == NULL ) {
        procerr( "cannot open log file\n");
    }
}

if( fileOfField[0] == '\0' )
    procerr("not specified file of field name\n");

loadfield();

if( fileOfTemplate[0] == '\0' )
    procerr("not specified template file\n");

loadTemplate();

if( fileOfText[0] == '\0' )
    procerr("not specified file of text\n");

if( writeto[0] == '\0' )
    procerr("not specified target name [tscmd.writeto]\n");

getFileOfText();

```

```

if( counter != initCounter ) {
    saveCounter();
}

return 0;
}

int getFileOfText() {
    struct _finddata_t ffbk;
    char filepath[ MAX_FILE_NAME ];
    int fp,nPos;

    fprintf( fpLog , "Split text file from [%s]\n" , fileOfText );

    strcpy( filepath , fileOfText );
    nPos = strlen( filepath )-1;
    while( nPos > 0 && filepath[nPos] != '\0' )
        nPos--;

    if( nPos > 0 )
        nPos++;

    fp = _findfirst( fileOfText,&ffbkl);
    if( fp < 0 )
        return -1;

    while( 1 ) {

        if( !(ffbkl.attrib & _A_SUBDIR) ) {
            strcpy( (filepath+nPos) , ffbkl.name );
            fprintf( fpLog , "read text file [%s]\n" , filepath );
            readtext( filepath );
        }

        if( _findnext(fp ,&ffbkl) == -1 )
            break;

        fflush( fpLog );
    }

    _findclose(fp);

    fprintf(fpLog,"done !!\n");
    if( uniqueValue ) {
        flushSort();
        fprintf(fpLog,"Total item(s) : %d\n",CountEntry(0) );
    } else {
        fprintf(fpLog,"Total generated file : %d\n",CountEntry(0));
    }

    if( fileOfLog[0] != '\0' ) {
        fclose( fpLog );
    }

    return 0;
}

```

```

int loadfield() {
    FILE *fp;
    char tmpWord[ MAX_STRING_LEN ], *cStr[ MAX_STRING_LEN ];
    int pos, fPos, n, i;

    if( (fp = fopen( fileOfField , "r" )) == NULL ) {
        sprintf(tmpWord , "Could not open file of field [%s]\n", fileOfField );
        procerr( tmpWord );
    }

    fPos = 0;
    while( fPos < MAX_FIELD ) {
        tmpWord[0] = '\0';
        pos = freadln( fp , tmpWord );
        if( pos < 0 )
            break;    /* EOF */

        if( pos != 0 ) {
            c = lstrchr( tmpWord , "/" ); // remove comment
            if( c ) {
                *c = '\0';
            }

            strAllTrim( tmpWord );

            if( strlen( tmpWord ) != 0 ) {
                if( tmpWord[0] == '.' ) { // dot command
                    if( lstrchr( tmpWord , ".field_delimiter" ) != NULL ) {
                        n = strlen( ".field_delimiter" );
                        strcpy( deliField , tmpWord+n );
                        strAllTrim( deliField );
                    } else

                    if( lstrchr( tmpWord , ".record_delimiter" ) != NULL ) {
                        n = strlen( ".record_delimiter" );
                        strcpy( deliRecord , tmpWord+n );
                        strAllTrim( deliRecord );
                    } else

                    if( lstrchr( tmpWord , ".counter" ) != NULL ) {
                        n = strlen( ".counter" );
                        strcpy( cStr , tmpWord+n );
                        strAllTrim( cStr );

                        if( cStr[0] == ':' ) {
                            strcpy( tmpWord , cStr+1 );
                            sizeOfCounter = atoi( getword( tmpWord, &n ) );
                            while( tmpWord[n] && !isspace( tmpWord[n] ) )
                                n++;
                            strcpy( cStr , tmpWord+n );
                            strAllTrim( cStr );
                        }

                        n = i = 0;
                        while( ( n = atChar( ',', cStr+i ) ) >= 0 ) {
                            i += (n+1);
                        }
                    }
                }
            }
        }
    }
}

```

```

        if( strlen( cStr+i ) > 0 ) {
            counter = initCounter = atol( cStr+i );
        }
    }
} else {
    parseField( (char *) tmpWord , (char *) field[fPos].name , (char *) field[fPos].flag ,
                &field[fPos].size , &field[fPos].start );

    if( strcmp( field[fPos].name , "counter" ) == 0 ) {
        procerr( "field name 'counter' has been prohibited\n" );
    }
    fPos++;
}
}
}

if( strlen( deliField ) == 0 )
    procerr("invalid field delimiter\n");

if( strlen( deliRecord ) == 0 )
    procerr("invalid record delimiter\n");

if( field[0].name[0] == '\0' )
    procerr("invalid defined field\n");

if( initCounter != 0 ) {
    strcpy( field[fPos++].name , "counter" );
}

strcpy( field[fPos].name , deliRecord );
totalField = fPos;
fclose( fp );
return 0;
}

// format field   field_name[:l | {s,l} , flag
//   l = length
//   s = start

```

```

void parseField( char *word , char *f_name , char *f_flag , int *f_size , int *f_start ) {
char tmpWord[ MAX_STRING_LEN ];
int pos,n,i;

    f_name[0] = f_flag[0] = '\0';
    *f_size = 0;
    *f_start = 0;

    pos = 0;
    while( word[pos] && word[pos] != ':' && word[pos] != ';' ) {
        f_name[pos] = tolower(word[pos]);
        pos++;
    }
}

```

```

f_name[pos] = '\0';

if( word[pos] && word[pos] == ',' ) {
    pos++;
    while( word[pos] && isspace( word[pos] ) ) pos++;

    if( word[pos] == '{' ) {
        pos++;
        n = 0;
        while( word[pos] && word[pos] != ',' && word[pos] != '}' ) {
            tmpWord[n++] = word[pos++];
        }
        tmpWord[n] = '\0';
        *f_start = atoi( tmpWord );

        if( word[pos] == ';' ) {
            pos++;
            n = 0;
            while( word[pos] && word[pos] != ',' && word[pos] != '}' ) {
                tmpWord[n++] = word[pos++];
            }
            tmpWord[n] = '\0';
            *f_size = atoi( tmpWord );

            while( word[pos] && word[pos] != ';' ) pos++;
        }
    } else {
        n = 0;
        while( word[pos] && word[pos] != ';' ) {
            tmpWord[n++] = word[pos++];
        }
        tmpWord[n] = '\0';
        *f_size = atoi( tmpWord );
    }
}
if( word[pos] && word[pos] == ';' ) {
    strcpy( f_flag, (char *) (word+pos+1) );
    strAllTrim( f_flag );
}
}
}

```

```

int loadTemplate() {
    FILE *fp;
    char *c, line[ MAX_STRING_LEN ], word[ MAX_STRING_LEN ];
    int pos,skiplen,fPos,i,n;

    if( (fp = fopen( fileOfTemplate, "r" )) == NULL ) {
        sprintf( line, "Could not open template file [%s]\n", fileOfTemplate );
        procerr( line );
    }

    while( 1 ) {
        pos = freadln( fp, line );
        if( pos < 0 )
            break; /* EOF */

        if( pos == 0 ) { /* new line */

```



```

        templateList = addToString( ID_STR_TEMPLATE , templateList , "\n" );
        continue;
    }

    c = lstrchr( line , "/" ); // remove comment
    if( c != NULL ) {
        *c = '\0';
        if( strlen(line) == 0 )
            continue;
    }
    if( !metaCMD(line) )
        templateList = addToString( ID_STR_TEMPLATE , templateList , line );
}
}

```

// parse content in meta HTML

// ex: <meta name="TSCMD.ccc" content="vvv">

```

int metaCMD( char *line ) {
int n,i,nPos;
char cmd[100],value[200] , tmpStr[500],tmpLine[ MAX_STRING_LEN ];

if( atlStr( "tscmd." , line ) < 0 ) { // no ts command in this buffer;
    return 0;
}
strcpy( tmpLine , line );
line[0] = '\0';
nPos = 0;
while( 1 ) {
    if( (i = atlStr( "tscmd." , tmpLine+nPos )) < 0 ) {
        strcat( line , (tmpLine+nPos) );
        break;
    }
    /* -- #1 read data before tag '<' */
    n = i;
    while( n >= 0 && tmpLine[nPos+n] != '<' ) {
        n--;
    }

    if( n < 0 ) { // err: no found '<'
        break;
    }

    if( n > 0 ) {
        tmpLine[nPos+n] = '\0';
        strcat( line , (tmpLine+nPos) );
        nPos += (n+1);
    }
    /* -- end of #1 -- */

    i = atlStr( "tscmd." , tmpLine+nPos );
    i += strlen( "tscmd." );
    n = 0;
    while( tmpLine[nPos+i] && tmpLine[nPos+i] != "" ) {
        cmd[n] = tolower( tmpLine[nPos+i] );
        i++;
        n++;
    }
}
}

```

```

    }
    cmd[n] = '\0';

    i = at1Str( "content", tmpLine+nPos );
    if( i < 0 )
        break;
    i += strlen("content");
    while( tmpLine[nPos+i] == '=' ||
           tmpLine[nPos+i] == '' ||
           tmpLine[nPos+i] == '\"' )
        i++;
    n = 0;
    while( tmpLine[nPos+i] && tmpLine[nPos+i] != '\"' ) {
        value[n] = tmpLine[nPos+i];
        i++;
        n++;
    }
    value[n] = '\0';
    /* -- #2 check tscmd.object -- */

    if( strcmp( cmd , "writeto" ) == 0 ) {
        strcpy( writeto , value );
    }

    /* -- end of #2 -- */
    while( tmpLine[nPos+i] && tmpLine[nPos+i] != '>' )
        i++;

    if( tmpLine[nPos+i] != '>' )
        break;

    nPos += (i+1);
}
return (strlen(line) != 0 ? 0 : 1);
}

int usage() {
    printf("Usage: gmain [-f file] [-p file] [-t file|dir ] [-l file] [-T file]\n");
    printf("\t-f : File contain field lists\n");
    printf("\t-p : Template file\n");
    printf("\t-t : Text file or dir path\n");
    printf("\t-l : print log to file\n");
    printf("\t-T : path of newdict.txt\n");
    exit(0);
}

void procerr(char *err ) {
    fprintf(fpLog,err);
    if( fileOfLog[0] != '\0' ) {
        fclose( fpLog );
    }
    exit(1);
}

```

```

int CountEntry(int n) {
static int nEntry = 0;
    if( n == 1 )
        nEntry++;
    return nEntry;
}

int saveCounter() {
    FILE *fp;
    char tmpWord[ MAX_STRING_LEN ], cStr[ MAX_STRING_LEN ];
    int pos,i;
    struct String *st,*tmpStr;

    st = NULL;

    if( (fp = fopen( fileOfField , "r+" )) == NULL ) {
        sprintf(tmpWord, "Could not open file of field [%s]\n", fileOfField );
        procerr( tmpWord );
    }

    while( TRUE ) {
        tmpWord[0] = '\0';
        pos = freadln( fp , tmpWord );
        if( pos < 0 )
            break; /* EOF */
        i = 0;
        while( tmpWord[i] && isspace( tmpWord[i] ) ) i++;

        if( atIstr( ".counter" , tmpWord+i ) == 0 ) {
            // ok right
            cStr[0] = '\0';
            xlStr( cStr , counter , sizeOfCounter );
            sprintf( tmpWord , "%s,%s", tmpWord,cStr );
        }
        st = addToString( 9 , st , tmpWord );
    }
    if( st != NULL ) {
        fseek( fp , 0 , 0 );
        tmpStr = st;
        while( tmpStr ) {
            fprintf( fp , "%s\n",tmpStr->line );
            tmpStr = tmpStr->next;
        }
    }
    fclose( fp );
    return 0;
}

```

### 1.2.6 gnmem.c จัดการเกี่ยวกับหน่วยความจำ

```

/* gnmem.c */
#include "gnconfig.h"
#include "gnmem.h"

void *emalloc(int i) {
    void *p;

    if ((p = (void *) malloc(i)) == NULL)
        procerr("Out of memory\n");
    return p;
}

void *erealloc(void *ptr, int i) {
    void *p;

    if ((p = (void *) realloc(ptr, i)) == NULL)
        procerr("Out of memory\n");
    return p;
}

char *memcpy( char *t , char *s , int start , int count , int i ) {

    if( start < 0 || start > strlen(s) )
        return NULL;
    if( count == 0 )
        count = strlen( s ) - start -1;
    while( count-- > 0 && s[start] ) {
        t[ i++ ] = s[ start++ ];
    }
    t[i] = '\0';
    return t;
}

```

### 1.2.7 gnread.c จัดการเกี่ยวกับการอ่านข้อมูลจากแฟ้ม

```

/* gnread.c */

#include "gnconfig.h"
#include "gnmem.h"

char *readblock( FILE * , char * );
int cmpFieldChar( char , char , int );
int cmpFieldStr( char* , char* , int );
int getFieldname( char * ,int );
int loadOldSort( int );

int getOldSort = 0;

// read block

void readtext(char *filename) {
    FILE *fp;
    char ch , *buffer , word[ MAX_STRING_LEN ] , cStr[ MAX_STRING_LEN ];
    char fileTarget[ MAX_FILE_NAME ];
    char *fieldValue[ MAX_FIELD ];

```

```

char f_name[ MAX_FIELD_LEN ],f_flag[ MAX_FLAG_LEN ];
int fieldPos[ MAX_FIELD ][2];
int i,j , fPos,mPos,nPos,lPos,lfPos,maxlen,deliPos, f_size , f_start;

buffer = (char *) emalloc( sizeof(char) * BUFFER_SIZE );

if( (fp = fopen( filename , "r" )) == NULL ) {
    sprintf( buffer , "Cannot open text file [%s]\n" , filename );
    procerr( buffer );
}
setCounter(0);
while( 1 ) {

    // initial field value
    i = 0;
    while( i < MAX_FIELD ) {
        fieldValue[i] = NULL;
        fieldPos[i][0] = i;
        fieldPos[i][1] = -1;
        i++;
    }
    if( setCounter(-1) > 0 ) {
        counter++;
    }
    setCounter(0);
    buffer = (char *) readblock( fp , buffer );
    if( buffer[0] == '\0' )
        break;
    maxlen = strlen(buffer)-1;

    // 1. check all field position
    // 2. sort field position
    // 3. process field by order 2.
    fPos = 0;
    while( fPos < totalField ) {
        mPos = 0;
        while( mPos < maxlen ) {
            nPos = atoiStr( field[ fPos ].name , (char *) (buffer+mPos) );
            if( nPos < 0 )
                break;
            lPos = mPos+nPos+strlen(field[fPos].name);
            while( buffer[lPos] && buffer[lPos] == ' ' ) {
                lPos++;
            }
            i = 0;
            while( deliField[i] && buffer[lPos] && deliField[i] == buffer[lPos] ) {
                i++;
                lPos++;
            }
            if( !deliField[i] ) {
                fieldPos[ fPos ][1] = mPos+nPos;
                break;
            }
            mPos += nPos+strlen(field[fPos].name);
        }
        fPos++;
    }
    fieldPos[ totalField ][1] = maxlen;

    // sort field pos : bubble sort

```

```

for( i=0; i < MAX_FIELD-1; i++ ) {
  for( j=i; j < MAX_FIELD; j++ ) {
    if( fieldPos[j][1] < fieldPos[i][1] ) {
      nPos = fieldPos[i][0];
      mPos = fieldPos[i][1];
      fieldPos[i][0] = fieldPos[j][0];
      fieldPos[i][1] = fieldPos[j][1];
      fieldPos[j][0] = nPos;
      fieldPos[j][1] = mPos;
    }
  }
}

nPos = 0;
while( nPos < MAX_FIELD && fieldPos[nPos][1] < 0 ) {
  nPos++;
}
mPos = 0;
fileTarget[0] = '\0';
while( nPos < MAX_FIELD-1 ) {
  fPos = fieldPos[nPos][0];
  mPos = fieldPos[nPos][1]+1;
  deliPos = atlStr( deliField, (char *) (buffer+mPos) );
  if( deliPos < 0 ) {
    break;
  }
  mPos += (deliPos+strlen( deliField ));
  while( buffer[mPos] == ' ' ||
        buffer[mPos] == '\n' ||
        buffer[mPos] == '\r' ) {
    mPos++;
  }
  buffer[ fieldPos[nPos+1][1] ] = '\0'; // cut to next field
  fieldValue[ fPos ] = (char *) (buffer+mPos);
  flushEOL( fPos, fieldValue[fPos] );
  nPos++;
}
// assigned new counter
if( initCounter != 0 ) {
  fPos = 0;
  while( fPos < MAX_FIELD-1 ) {
    if( strcmp( field[ fPos ].name, "counter" ) == 0 ) {
      xlStr( cStr, counter, sizeofCounter );
      fieldValue[fPos] = cStr;
      break;
    }
    fPos++;
  }
}
// get writeto name
nPos = mPos = 0;
while( writeto[ nPos ] ) {
  if( writeto[ nPos ] == '%' ) {
    nPos++;
    j = 0;
    while( writeto[nPos] && writeto[nPos] != '%' ) {
      word[j++] = tolower(writeto[nPos++]);
    }
    if( writeto[nPos] == '%' ) { // ok this field

```

```

        word[j] = '\0';
        nPos++;
        fileTarget[mPos] = '\0';
        parseField( word , f_name , f_flag , &f_size , &f_start);
        getFieldValue( fileTarget , fieldValue , f_name , f_flag , f_size , f_start);
        mPos = strlen( fileTarget );
    }
} else {
    fileTarget[mPos++] = writeto[nPos++];
}
}
fileTarget[mPos] = '\0';
if( strlen( fileTarget ) > MAX_FILE_NAME-2 ) {
    sprintf( cStr , "name of file target too long [%s]\n" , fileTarget );
    procerr( cStr );
}
if( uniqueValue ) {
    throwToSort( fieldValue );
} else if( fileTarget[0] != '\0' ) {
    if( throwToTemplate( fileTarget , fieldValue ) < 0 )
        break;
}
}
fclose(fp);
free( buffer );
}

```

```

int flushEOL( int fPos , char *fieldValue ) {
    int n,i;
    n = strlen( fieldValue )-1;
    // flush EOL or space
    while( n > 0 ) {
        if( fieldValue[n] != ' ' && fieldValue[n] != '\n' )
            break;
        n--;
    }
    fieldValue[n+1] = '\0';
}

```

```

// get rid EOL with in field
if( atChar( 'N' , field[fPos].flag ) < 0 )
    return 0;

```

```

n = 0;
while( fieldValue[n] ) {
    if( fieldValue[n] == '\n' )
        fieldValue[n] = ' ';
    n++;
}
return 1;
}
}

```

```

#define SIZE_READ 8192

```

```

char *readblock( FILE *fp , char *buffer ) {
    int nRead,ch,endPoint,n;
    nRead = endPoint = 0;

    buffer[0] = '\0';
    while( 1 ) {
        ch = fgetc( fp );

```

```

if( ch == EOF )
    break;
ch = (unsigned char) ch;

/* check blank at first read */
if( nRead == 0 ) {
    if( ch == ' ' || ch == '\n' )
        continue;
}
/* is end of block */
if( field[totalField].name[ endPoint ] == '\0' ) {
    /* read while EOL */
    if( ch != '\n' ) {
        continue;
    }

    /* OK exit */
    break;
}
if( cmpFieldChar( ch , field[ totalField ].name[ endPoint ] , totalField ) ) {
    endPoint++;
    continue;
}
// split invalid field
if( endPoint > 0 ) {
    n = 0;
    while( n < endPoint ) {
        buffer[ nRead++ ] = field[ totalField ].name[n++];
    }
}
endPoint = 0; // don't end of block , new checked
buffer[nRead++] = ch;
}
buffer[nRead] = '\0';
return buffer;
}

```

```

int cmpFieldChar( char ch , char fch , int fPos ) {
    /* not case sensitive */
    if( atChar( 'U' , field[ fPos ].flag ) >= 0 )
        return (tolower(ch) == tolower(fch));
    return (ch == fch);
}

```

```

int getfieldname( char *buffer , int nPos ) {
    char savech;
    char cfield[100];
    int i,n;

    savech = buffer[nPos];
    buffer[nPos] = '\0';
    i = nPos-1;
    while( i > 0 && buffer[i] != '\n' && buffer[i] != ' ' ) {
        i--;
    }
    memcpy( cfield , buffer , ++i , nPos-i , 0 );
    buffer[ nPos ] = savech;
    n = 0;
    while( n < MAX_FIELD ) {

```



```

    if( nstrchr( field[n].name , cfield ) != NULL )
        break;
    n++;
}
return ( n < MAX_FIELD ? n : -1);
}

int throwToTemplate( char *fileTarget , char *fieldValue[] ) {
    struct String *tmpLst;
    char cStr[ MAX_STRING_LEN ],cField[ MAX_FIELD_LEN ];
    char f_name[ MAX_FIELD_LEN ],f_flag[ MAX_FLAG_LEN ];
    char *tmpStr,*tmpWord,*saveWord;
    FILE *fpTarget;
    int i,j,nPos,fPos,n,r,newline,f_size,f_start;

    if( (fpTarget = fopen( fileTarget , "w" )) == NULL ) {
        sprintf( cStr , "Cannot create file [%s]\n",fileTarget);
        perror( cStr );
    }

    fprintf(fpLog,"tMake target [%s]\n", fileTarget );

    tmpLst = (struct String *) templateList;
    tmpStr = (char *) emalloc( sizeof( char )* BUFFER_SIZE );
    tmpWord = (char *) emalloc( sizeof( char )* MAX_STRING_LEN );
    saveWord = (char *) emalloc( sizeof( char )* MAX_STRING_LEN );

    while( tmpLst != NULL ) {
        strcpy( cStr , tmpLst->line );
        nPos = 0;
        if( emptyStr( cStr ) ) {
            nPos = -1;
        }
        while( nPos >= 0 ) {
            i = atChar( '%', cStr );
            if( i < 0 ) {
                fputs( cStr , fpTarget );
                nPos = -1;
            } else {
                j = atChar( '%', (char *) (cStr+i+1) );
                if( j < 0 ) {
                    fputs( cStr , fpTarget );
                    nPos = -1;
                } else {
                    if( i > 0 ) {
                        cStr[i] = '\0';
                        fputs( cStr , fpTarget ); // print front of Template
                    }
                    i++;
                    j = 0;
                    while( cStr[i] != '%' )
                        cField[j++] = cStr[i++];
                    cField[j] = '\0';
                    nPos += i+1;
                    parseField( cField , f_name , f_flag , &f_size , &f_start );
                    fPos = getFieldPos( f_name );

                    if( fPos < 0 ) {
                        nPos--;
                        fprintf( fpTarget , "%%%" , cField );
                    }
                }
            }
        }
    }
}

```

```

} else if( fieldValue[fPos] == NULL ) {

    // nothing

} else {

    // assigned default if empty
    if( strlen( f_flag ) == 0 )
        strcpy( f_flag , field[fPos].flag );

    if( f_size == 0 )
        f_size = field[fPos].size;

    if( f_start == 0 )
        f_start = field[fPos].start;

    n = -1;
    if( atoiChar( 'T' , f_flag ) >= 0 ) {
        n = 0; // t or T
    }

    if( n < 0 ) { // no Thai formatted
        tmpStr[0] = '\0';
        getFieldValue( tmpStr , fieldValue , f_name , f_flag , f_size , f_start );
        fputs( tmpStr , fpTarget );
    } else {
        saveWord[0] = '\0';
        n = 0;
        while( fieldValue[fPos][n] ) {
            // check newline & space
            while( 1 ) {
                if( fieldValue[fPos][n] == '\n' ||
                    fieldValue[fPos][n] == '\r' ) {
                    fputs( "\n" , fpTarget );
                    break;
                }
                if( !isspace( fieldValue[fPos][n] ) )
                    break;
                n++;
            }

            r = 0;
            if( saveWord[0] != '\0' ) {
                strcpy( tmpWord , saveWord );
                r = strlen( tmpWord );
                saveWord[0] = '\0';
            }

            while( fieldValue[fPos][n] &&
                !isspace( fieldValue[fPos][n] ) &&
                fieldValue[fPos][n] != '\n' &&
                fieldValue[fPos][n] != '\r' ) {
                tmpWord[r++] = (unsigned char) fieldValue[fPos][n++];
            }
            tmpWord[r] = '\0';

            newline = (fieldValue[fPos][n] == '\n' || fieldValue[fPos][n] == '\r');

```

```

        if( newline )
            n++;

        if( strlen(tmpWord) == 0 ) {
            continue;
        }
        strcat( tmpWord , " " );
        thaiword( tmpStr , tmpWord );
        if( newline ) {
            // save last word
            r = strlen(tmpStr)-1;
            while( r > 0 && tmpStr[r] == ' ' ) {
                tmpStr[r--] = '\0';
            }
            while( r > 0 && tmpStr[r] && tmpStr[r] != ' ' ) {
                r--;
            }
            if( tmpStr[r] == ' ' ) {
                tmpStr[r] = '\0';
                strcpy( saveWord , (char *) (tmpStr+r+1) );
            }
        }
        strcat( tmpStr , " " );
        if( newline )
            strcat( tmpStr , "\n" );

        fputs( tmpStr ,fpTarget );
    }
}

if( nPos >= strlen( tmpLst->line ) )
    break;
strcpy( cStr , tmpLst->line+nPos );
}
}
}
fprintf( fpTarget , "\n" );
tmpLst = (struct String *) tmpLst->next;
}

```

```

free( tmpStr );
free( tmpWord );
free( saveWord );

fclose( fpTarget );
CountEntry(1);
return 0;
}

```

```

int emptyStr( char *str ) {
    int i=0;
    while( str[i] ) {
        if( str[i] != ' ' && str[i] != '\n' && str[i] != 9 )
            return 0;
        i++;
    }
    return 1;
}

```

```

int throwToSort( char *fieldValue[] ) {
    struct String *tmpLst;
    struct sortedString *tmpSort;
    char cStr[ MAX_STRING_LEN ], cField[ MAX_FIELD_LEN ];
    char *tmpStr;
    int i,j,nPos,fPos;

    tmpLst = (struct String *) templateList;
    tmpStr = (char *) emalloc( sizeof( char ) * BUFFER_SIZE );
    while( tmpLst != NULL ) {
        strcpy( cStr , tmpLst->line );
        nPos = 0;
        while( nPos >= 0 ) {
            i = atChar( '%', cStr );
            if( i < 0 ) {
                nPos = -1;
            } else {
                j = atChar( '%', (char *) (cStr+i+1) );
                if( j < 0 ) {
                    nPos = -1;
                } else {
                    cStr[i] = '\0';
                    i++;
                    j = 0;
                    while( cStr[i] != '%' )
                        cField[j++] = cStr[i++];
                    cField[j] = '\0';
                    nPos += i+1;
                    fPos = 0;
                    while( fPos < MAX_FIELD ) {
                        if( field[fPos].name[0] == '\0' ||
                            strcmp( field[fPos].name , cField ) == 0 )
                            break;
                        fPos++;
                    }
                    if( fPos < MAX_FIELD && field[fPos].name[0] != '\0' ) {
                        if( getOldSort == 0 ) {
                            loadOldSort( fPos );
                            getOldSort = 1;
                        }
                        // Ok is field
                        if( fieldValue[fPos] != NULL ) {
                            tmpSort = (struct sortedString *) newSort( fieldValue[fPos] );
                            if( fieldSort[fPos] == NULL )
                                fieldSort[fPos] = tmpSort;
                            else
                                addSort( fieldSort[fPos] , tmpSort );
                        }
                    }
                    if( nPos >= strlen( tmpLst->line ) )
                        break;
                    strcpy( cStr , tmpLst->line+nPos );
                }
            }
            tmpLst = (struct String *) tmpLst->next;
        }
        CountEntry(1);
        return 0;
    }
}

```

```

int flushSort() {
    FILE *fp = NULL;
    int nPos;
    if( fileOfUnique[0] != '\0' )
        fp = (FILE*) fopen( fileOfUnique , "w" );
    for( nPos = 0; nPos < MAX_FIELD ; nPos++ ) {
        if( fieldSort[nPos] != NULL )
            printSort( fp , fieldSort[nPos] );
    }
    if( fp != NULL )
        fclose( (FILE*) fp );
    return 0;
}

int printSort( FILE *fp , struct sortedString *aSort ) {
    if( aSort->left != NULL )
        printSort( fp , aSort->left );
    if( fp != NULL )
        fprintf( fp , "%s\n",aSort->line );
    //fprintf(fpLog, "%s\n",aSort->line );
    if( aSort->right != NULL )
        printSort( fp ,aSort->right );
}

int loadOldSort( int fPos ) {
    FILE *fp = NULL;
    char cStr[ MAX_STRING_LEN ];
    char ch;
    int nPos;
    struct sortedString *tmpSort;

    if( fileOfUnique[0] == '\0' )
        return 0;

    if( (fp = fopen( fileOfUnique , "r" )) == NULL )
        return 0;

    nPos = 0;
    while(1) {
        ch = fgetc( fp );
        if( ch == EOF || ch == '\n' ) {
            cStr[nPos] = '\0';
            if( nPos > 0 ) {
                tmpSort = (struct sortedString *) newSort( cStr );
                if( fieldSort[fPos] == NULL )
                    fieldSort[fPos] = tmpSort;
                else
                    addSort( fieldSort[fPos] , tmpSort );
            }
            if( ch == EOF )
                break;
            nPos = 0;
        } else {
            cStr[nPos++] = (char) ch;
        }
    }
    fclose( fp );
    return 0;
}

```

```

// Check counter was used ?
int setCounter(int newSet ) {
static int nCountSet = 0;
    if( newSet > 0 ) {
        nCountSet = newSet;
    }
    return nCountSet;
}

int getFieldPos( char *f_name ) {
int fPos , j , l;
    if( strcmp( f_name , "counter" ) == 0 ) {
        setCounter(1);
    }
    fPos = 0;
    while( fPos < totalField && strcmp( field[fPos].name , f_name ) != 0 )
        fPos++;
    return ( fPos == totalField ? -1 : fPos);        // not found or empty
}

int getFieldValue( char *buffer , char *fieldValue[] , char *f_name , char *f_flag , int f_size , int f_start ) {
int fPos , j , l;

    fPos = getFieldPos( f_name );
    if( fPos < 0 || fieldValue[ fPos ] == NULL )        // not found or empty
        return 0;

    if( f_size == 0 )
        f_size = field[fPos].size;

    if( f_start == 0 )
        f_start = field[fPos].start;

    if( f_size < 0 && f_start > 0 )
        f_size = abs(f_size);

    if( f_size == 0 ) {
        if( f_start < strlen(fieldValue[fPos])-1 )
            strcat( buffer , (char *) (fieldValue[fPos]+f_start) );
        return 1;
    }
    l = strlen( buffer );
    if( f_size > 0 ) {
        j = f_start;
        while( fieldValue[fPos][j] && j < f_size ) {
            buffer[l++] = fieldValue[fPos][j++];
        }
        buffer[l] = '\0';
    } else {
        j = strlen( fieldValue[fPos] );
        if( j >= abs(f_size) ) {
            j = abs(f_size);
            while( fieldValue[fPos][j] ) {
                buffer[l++] = fieldValue[fPos][j++];
            }
            buffer[l] = '\0';
        }
    }
    return 1;
}

```

## 1.2.8 gnstring.c จัดการเกี่ยวกับค่าสายอักขระ

```

/* gnstring.c */
#include "gnconfig.h"

struct String *addtoString( unsigned char nID , struct String *st , char *s ) {
    static struct String *saveNode;
    struct String *tmpNode;

    tmpNode = (struct String *) emalloc( sizeof(struct String) );
    strcpy(tmpNode->line,s);
    tmpNode->id = nID;

    if( st == NULL )
        st = tmpNode;
    else {
        if( saveNode != NULL && saveNode->id != nID ) {
            saveNode = st;
            while( saveNode->next != NULL )
                saveNode = saveNode->next;
        }
        saveNode->next = tmpNode;
    }
    saveNode = tmpNode;
    return st;
}

char *strAllTrim( char *buffer ) {
    char s[ MAX_STRING_LEN ];
    int i;
    strcpy( s , buffer );
    for( i=0; s[i] && s[i] == ' '; i++ )
        ;
    strcpy( buffer , (s+i) );
    i = strlen(buffer)-1;
    while( i > 0 && buffer[i] == ' ' )
        i--;
    buffer[i+1] = '\0';
    return buffer;
}

char *strLower( char *buffer ) {
    int i;
    for( i=0; buffer[i]; i++ )
        buffer[i] = tolower( buffer[i] );
    return buffer;
}

char tolower(char c) {
    char i;
    i = c;
    if( i >= 'A' && i <= 'Z' ) {
        i = 'a'+( i-'A' );
    }
    return i;
}

```

```
int isspace(int c) {
    return ( c == 9 || c == 32 || c == 160 );
}
```

```
/* inspect is 't' in 's' */
int atStr(char *t, char *s) {
    int i, j, k;
    i = 0;
    while( s[i] ) {
        if( t[0] == s[i] ) {
            j = i;
            k = 0;
            while( s[j] && t[k] && s[j] == t[k] ) {
                j++;
                k++;
            }
            if( !t[k] )
                return i;
        }
        i++;
    }
    return -1;
}
```

```
/* inspect is 't' in 's' with not case sensitive */
int atlStr(char *t, char *s) {
    int i, j, k;
    i = 0;
    while( s[i] ) {
        if( tolower( t[0] ) == tolower( s[i] ) ) {
            j = i; k = 0;
            while( s[j] && t[k] && tolower( s[j] ) == tolower( t[k] ) ) {
                j++; k++;
            }
            if( !t[k] )
                break;
        }
        i++;
    }
    if( s[i] ) return i;
    return -1;
}
```

```
int atChar( char t , char *s) {
    int i = 0;
    while( s[i] && t != s[i] )
        i++;
    if( !s[i] )
        return -1;
    return i;
}
```

```
int atlChar( char t , char *s) {
    int i = 0;
    while( s[i] && tolower(t) != tolower(s[i]) )
        i++;
    if( !s[i] ) return -1;
    return i;
}
```



```

int ratChar( char t , char *s , int skip ) {
    int i;
    i = strlen(s)-1-skip;
    while( i >= 0 && s[i] != t )
        i--;
    return i;
}

// check t is subset of s
char *lstrstr(char *s, char *t) {
    int i, j, k, l;
    for (i = 0; s[i]; i++) {
        for (j = 0, l = k = i; s[k] && t[j] && tolower(s[k]) == tolower(t[j]); j++, k++);
        if (t[j] == '\0')
            return s + l;
    }
    return NULL;
}

char *nstrstr(char *s, char *t) {
    int i, j, k, l;
    for (i = 0; s[i]; i++) {
        for (j = 0, l = k = i; s[k] && t[j] && s[k] == t[j]; j++, k++);
        if (t[j] == '\0')
            return s + l;
    }
    return NULL;
}

/* Gets the next word in a line. If the word's in quotes,
** include blank spaces in the word or phrase.
*/

char *getword(char *line, int *skiplen) {
    int i;
    char *start;
    static char word[ MAX_STRING_LEN ];

    start = line;
    if (!(*line))
        return "\0";

    while (isspace(*line))
        line++;

    if (!(*line))
        return "\0";

    for (i=0; *line && !isspace(*line) && i < MAX_STRING_LEN ; line++)
        word[i++] = *line;

    word[i] = '\0';

    *skiplen = line - start;

    return word;
}

```

```

int sortEntry( struct sortedString *aSort , struct String *Str ) {
    struct sortedString *aSorttmp;
    struct String *tmpStr;
    tmpStr = (struct String *) Str;
    while( tmpStr ) {
        aSorttmp = (struct sortedString *) newSort( tmpStr->line );
        if( aSort == NULL )
            aSort = (struct sortedString *) aSorttmp;
        else
            addSort( aSort , aSorttmp );
        tmpStr = tmpStr->next;
    }
    return 0;
}

struct sortedString *newSort(char *cLine) {
    struct sortedString *tmp;
    tmp = (struct sortedString *) emalloc( sizeof(struct sortedString) );
    tmp->left = tmp->right = NULL;
    strcpy( tmp->line , cLine );
    strLower( tmp->line );
    return tmp;
}

int addSort( struct sortedString *aSort , struct sortedString *aSorttmp ) {
    int n;
    if( aSort == NULL ) {
        aSort = aSorttmp;
        return 2;
    }
    n = strcmp( aSorttmp->line , aSort->line );
    if( n == 0 ) {
        free( aSorttmp );
        return 0;
    }
    if( n < 0 ) {
        if( aSort->left == NULL ) {
            aSort->left = aSorttmp;
            return -1;
        }
        return addSort( aSort->left , aSorttmp );
    } else {
        if( aSort->right == NULL ) {
            aSort->right = aSorttmp;
            return 1;
        }
        return addSort( aSort->right , aSorttmp );
    }
    return 0;
}

int freeSort( struct sortedString *aSort ) {
    if( aSort->left != NULL )
        freeSort( aSort->left );
    if( aSort->right != NULL )
        freeSort( aSort->right );
    free(aSort);
    aSort = NULL;
    return 0;
}

```

```
// convert long number to string replace space with 0
int xlStr( char *buffer , long nNum , int nSize ) {
    char nb[40];
    int i;
    ltoa( nNum , nb , 10 );
    strAllTrim( nb );
    if( nSize < strlen(nb) ) {
        for( i=0; i < nSize; i++ ) {
            buffer[i] = '*';
        }
    } else {
        for( i=0; i < (nSize-strlen(nb)); i++ ) {
            buffer[i] = '0';
        }
        buffer[i] = '\0';
        strcat( buffer , nb );
    }
    return 1;
}
}
```

### 1.2.9 thaiword.c

ฟังก์ชันจัดทำดัชนีคำ

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
#include <float.h>
#include <sys/timeb.h>

#define _THAI_WORD_
#define _GNMAIN_

#include "thaiword.h"

#define DIC_MAX 56500
#define KEYWORD_MAX 30
#define INPUT_MAX 200

typedef struct {
    char word[KEYWORD_MAX];
} dic_type;

dic_type dic[DIC_MAX];

typedef struct {
    unsigned int start;
    unsigned int end;
    unsigned int dickey;
    unsigned int ch[INPUT_MAX];
} newcheck_type;

typedef struct {
    unsigned int num;
    unsigned int sum;
    unsigned list[KEYWORD_MAX];
} path_type;
```

```

void initialize( char *);
int indic(char *);
void thaiword(char *Result ,char *Source );
void sepword(char*,char *);
void shortpath(unsigned int,unsigned int);
void thaisep();
unsigned char *FINDCUT();

unsigned int dic_key=0;
unsigned int keyword_key=0;
int verbose = 0;
char *dictName = NULL;

unsigned int list[INPUT_MAX/2][INPUT_MAX/2]; // use in separate15
path_type path[2]; // use in
separate15

int init_Ready = 0;
FILE *fp_thaikey = NULL;

/* Diction Section */

void initialize( char *fname ) {
    getdic( fname );

    //openthaikey();
    init_Ready = 1;
}

/* ----- End of Dictionary Initial ----- */

/*
cTarget must be allocated
*/

void thaiword(char *cTarget , char *cSource ) {
    char str[4096];
    int i,n;
    char ch;

    if( !init_Ready ) {
        initialize( "NewDict.txt" );
    }

    cTarget[0] = '\0';
    n = i=0;
    while( cSource[n] ) {
        ch=cSource[n++];
        if((((ch>='n')&&(ch<=';'))||((ch>='i')&&(ch<=''))&&(ch!='\')&&(ch!='\r')) {
            str[i++]=ch;
        } else {
            if(i>0) {
                str[i]='\0';
                sepword( cTarget , str);
            }
            sprintf( cTarget , "%s%c",cTarget,ch );
            i=0;
        }
    }
    if(i>1) {

```

```

    str[i]='\0';
    sepword( cTarget , str);
}

}

void shortpath(unsigned int num,unsigned int pos) {
unsigned int i;

    if(pos>=num-1) {
        if(path[1].sum<=path[0].sum || path[0].sum==0) {
            path[0]=path[1];
        }
    } else {
        if(list[pos][pos+1]==0) {
            path[1].list[path[1].num++]=pos;
            shortpath(num,pos+1);
            path[1].num--;
        } else {
            i=1;
            while(list[pos][pos+i]>0 && pos+i<num) {
                path[1].sum=path[1].sum+list[pos][pos+i];
                path[1].list[path[1].num++]=pos;
                shortpath(num,pos+i);
                path[1].num--;
                path[1].sum=path[1].sum-list[pos][pos+i];
                i++;
            }
        }
    }
}

/*
get thaiword || after word must have 2 spaces
*/

void sepword(char *cTarget , char *str) {
newcheck_type check[INPUT_MAX],check1[INPUT_MAX],check2[INPUT_MAX];
unsigned int num=0;
unsigned int num1=0;
unsigned int num2=0;
unsigned int i,j,k;
char temp[INPUT_MAX];
int p;
int sep[INPUT_MAX],sep1[INPUT_MAX];
unsigned int xxx;
int chk[INPUT_MAX]; // ADD Prevent Recursive

/* edited by stk */
int oldPos;
for(i=0;i<strlen(str)-1;i++) {
    j=strlen(str)-1;
    while(j>i) {
        if(j-i+1<=keyword_key) {
            strcpy(temp,str+i);
            temp[j-i+1]='\0';
            if((p=indic(temp))!= 0) {
                if(num==0) {

```

```

        check[num].start=i;
        check[num].end=j;

        for(k=0;k<INPUT_MAX;k++)
            check[num].ch[k]=0;

        check[num].ch[i]=1;
        check[num].ch[j]=2;
        check[num++].dickey=p;

    } else {
        if(j>check[num-1].end) {
            check[num].start=i;
            check[num].end=j;
            for(k=0;k<INPUT_MAX;k++)
                check[num].ch[k]=0;
            check[num].ch[i]=1;
            check[num].ch[j]=2;
            check[num++].dickey=p;
        } else {
            if(i==check[num-1].start || j==check[num-1].end) {
                if(check[num-1].ch[i]%2==0)
                    check[num-1].ch[i]++;
                if((_lrotr(check[num-1].ch[j],1)%2)==0)
                    check[num-1].ch[j]+=2;
            } else {
                if((_lrotr(check[num-1].ch[i],2)%2)==0)
                    check[num-1].ch[i]+=4;
                if(check[num-1].ch[j]<8)
                    check[num-1].ch[j]+=8;
            }
        }
    }
}
j--;
}
}

if(num==0) {
    if( cTarget[0] == '\0' )
        sprintf(cTarget,"%s ",str);
    else
        sprintf(cTarget,"%s%s ",cTarget,str);

    return;
}

for(i=0;i<num;i++)
    for(j=0;j<num;j++)
        list[i][j]=0;

for(i=0;i<num-1;i++) {
    j=i+1;
    while(j<num && check[j].start<=check[i].end+1) {
        if(check[j].start==check[i].end+1) {
            list[i][j]=1;
        } else {
            if(_lrotr(check[i].ch[check[j].start-1],1)%2 && check[j].ch[check[i].end+1]%2) {
                list[i][j]=10;
            }
        }
    }
}
// weight

```

```

    } else {
        if(!_rotr(check[i].ch[check[j].start-1],1)%2 || check[j].ch[check[i].end+1]%2) {
            list[i][j]=100;
        } else {
            list[i][j]=1000;
        }
    }
}
j++;
}
}

```

```

path[0].num=path[1].num=0;
path[0].sum=path[1].sum=0;
shortpath(num,0); // find shorted path
path[0].list[path[0].num++]=num-1; // add last word in list

```

```

i=0;
if(i<path[0].num && check[path[0].list[i]].start>0) {
    check1[num1].start=0;
    check1[num1].end=check[path[0].list[i]].start-1;
    for(k=0;k<INPUT_MAX;k++)
        check1[num1].ch[k]=0;
    check1[num1].ch[check1[num1].start]=1;
    check1[num1].ch[check1[num1].end]=2;
    check1[num1++].dickey=dic_key;
}

```

```

check1[num1++]=check[path[0].list[i++]];
while(i<path[0].num) {
    if(check[path[0].list[i]].start>check1[num1-1].end+1) {
        check1[num1].start=check1[num1-1].end+1;
        check1[num1].end=check[path[0].list[i]].start-1;
        for(k=0;k<INPUT_MAX;k++)
            check1[num1].ch[k]=0;
        check1[num1].ch[check1[num1].start]=1;
        check1[num1].ch[check1[num1].end]=2;
        check1[num1++].dickey=dic_key;
    }
}

```

```

check1[num1++]=check[path[0].list[i++]];
}
if(check1[num1-1].end<strlen(str)-1) {
    check1[num1].start=check1[num1-1].end+1;
    check1[num1].end=strlen(str)-1;
    for(k=0;k<INPUT_MAX;k++)
        check1[num1].ch[k]=0;
    check1[num1].ch[check1[num1].start]=1;
    check1[num1].ch[check1[num1].end]=2;
    check1[num1++].dickey=dic_key;
}

```

```

for(i=0;i<INPUT_MAX;i++) {
    sep[i]=0;
    sep1[i]=0;
}

```

```

num=0;
i=0;

```

```

if(check1[i].dickey==dic_key) {
    for(j=check1[i].start;j<=check1[i].end;j++)
        sep[j]=-1;
    i++;
}
while(i<num1-1) {
    if(check1[i+1].dickey<dic_key) {
        if(check1[i+1].start==check1[i].end+1) {
            if(sep[check1[i].end]==0)
                sep[check1[i].end]=1;           // end word
        } else
            if(!_lrotr(check1[i].ch[check1[i+1].start-1],1)%2 && check1[i+1].ch[check1
[i].end+1]%2) {

                check[num].start=check1[i].start;
                check[num].end=check1[i+1].start-1;
                check[num++].dickey=0;
                check[num].start=check1[i].end+1;
                check[num].end=check1[i+1].end;
                check[num++].dickey=0;

                sep[check1[i].end]=1;           // end word
                sep[check1[i+1].start-1]=1;
                sep[check1[i+1].end]=1;
            } else {
                if(!_lrotr(check1[i].ch[check1[i+1].start-1],1)%2) {
                    check[num].start=check1[i].start;
                    check[num].end=check1[i+1].start-1;
                    check[num++].dickey=0;

                    sep[check1[i].end]=1;           // end word
                    sep[check1[i+1].start-1]=1;
                    sep[check1[i+1].end]=1;
                    j=check1[i+1].start;
                    k=0;
                    while(j<check1[i].end) {
                        if(!_lrotr(check1[i].ch[j],1)%2 && check1[i+1].ch[j+1]%2) {
                            k++;
                            check[num].start=check1[i].start;
                            check[num].end=j;
                            check[num++].dickey=0;
                            check[num].start=j+1;
                            check[num].end=check1[i+1].end;
                            check[num++].dickey=0;
                            sep[j]=1;           // end word
                        }
                        j++;
                    }
                } else {
                    if(check1[i+1].ch[check1[i].end+1]%2) {
                        check[num].start=check1[i].end+1;
                        check[num].end=check1[i+1].end;
                        check[num++].dickey=0;

                        sep[check1[i].end]=1;           // end word
                        sep[check1[i+1].end]=1;
                        j=check1[i+1].start;
                        k=0;
                        while(j<check1[i].end) {
                            if(!_lrotr(check1[i].ch[j],1)%2 && check1[i+1].ch[j+1]%2) {
                                k++;

```



```

        check[num].start=check1[i].start;
        check[num].end=j;
        check[num++].dickey=0;
        check[num].start=j+1;
        check[num].end=check1[i+1].end;
        check[num++].dickey=0;
        sep[j]=1; // end word
    }
    j++;
} else {
    j=check1[i+1].start;
    k=0;
    while(j<check1[i].end) {
        if(!_rotr(check1[i].ch[j],1)%2 && check1[i+1].ch[j+1]%2) {
            k++;
            check[num].start=check1[i].start;
            check[num].end=j;
            check[num++].dickey=0;
            check[num].start=j+1;
            check[num].end=check1[i+1].end;
            check[num++].dickey=0;
            sep[j]=1; // end word
            sep[check1[i+1].end]=1;
        }
        j++;
    }
    if(k==0)
        for(j=check1[i].start;j<=check1[i+1].end;j++)
            sep[j]=-1;
}
} else {
    if(sep[check1[i].end]==0)
        sep[check1[i].end]=1;
    for(j=check1[i+1].start;j<=check1[i+1].end;j++)
        sep[j]=-1;
    i++;
}
i++;
}

xxx=num1;

strcpy(temp,str);
thaisep(temp,sep1);
num1=xxx;

// new aDD_1
for(i=0;i<INPUT_MAX;i++) // ADD Prevent Recursive
    chk[i]=0;
// new aDD_1

i=0;
while(i<strlen(str)) {
    if(sep[i]==-1) {
        if(i>0) {
            j=i-1;

```

```

        while(sep1[j]==0 && j>0)
            j--;
        if(sep1[j]==1)
            j++;
    } else
        j=0;
    k=i;
    while(sep1[k]==0 && k<strlen(str))
        k++;
    while(check2[num2-1].start>=j && num2>0)
        num2--;

    //check2[num2].start=j;
    if(num2>0)
        check2[num2].start=check2[num2-1].end+1;
    else
        check2[num2].start=j;
    check2[num2++].end=k;

    if(chk[check2[num2-1].start]==0) { // ADD Prevent Recursive
        i=check2[num2-1].start;
        chk[i]=1;
    }
    j=0;

    while(check1[j].start<i && j<num1)
        j++;

    if(check1[j].start==i && j<num1 && check1[j].end-check1[j].start>4) {
        check2[num2].start=check1[j].end+1;
        check2[num2-1].end=check1[j].end;
        num2++;
        i=k;
        if(check2[num2-1].start>=check2[num2-1].end) {
            num2=2;
            for(j=check2[num2].start;j<check2[num2].end;j++)
                sep[j]=0;
            i=check2[num2].start-1;
        }
    } else
        i=k;
} else {
    j=0;
    while(check[j].start<i && j<num)
        j++;
    if(check[j].start==i && j<num) {
        check2[num2].start=i;
        check2[num2++].end=check[j].end;
    }
    j=0;
    while(check1[j].start<i && j<num1)
        j++;
    if(check1[j].start==i && j<num1) {
        check2[num2].start=i;
        check2[num2++].end=check1[j].end;
    }
}
}
i++;
}
}

```

```

// printf("str = [%s]\n",str);
oldPos = -1;

for(i=0;i<num2;i++) {

    // if( check2[i].start != oldPos ) {

    if(check2[i].end>=check2[i].start) {
        strcpy(temp,str+check2[i].start );
        temp[check2[i].end-check2[i].start+1]='\0';
        if( cTarget[0] == '\0' )
            sprintf(cTarget,"%s ",temp);
        else
            sprintf(cTarget,"%s %s ",cTarget,temp);

        oldPos = check2[i].start;
    }
}

void thaisep( input,sep1 )
unsigned char *input;
int sep1[];
{
    unsigned char *temp;
    int len,i;

    len=0;
    while(input[len]!='\0')
        len++;
    sep1[len-1]=1;
    i=len-1;
    while(i>0) {
        temp=FINDCUT(input,input+len,input+i);
        sep1[temp-input]=1;
        i=temp-input-1;
    }
}

/*-----
Module name : SYL_SEP (Syllable Separation)
Parameters : lm (Left Limit to search for cut point)
             eot (End Of Text)
             rm (Right Margin)
Function    : Find the location of the given text to be separated
Return     : Pointer to that location (Real Right Margin)

Author      : Tanat Chirakarn
             Subun Yenjaichon

Date Written : Friday, 6 May, 1988 8:38:18 AM
Last Update  : Tuesday, 19 May, 1988 10:07:31 AM
-----*/

/* The Main Function Thai Syllable Separate that will return
the Real Right Margin */

/* Function name must be capital letter for asm call */

```

```

unsigned char *FINDCUT(lm, eot, rm)
  unsigned char *lm, *eot, *rm; /* receive value from assembly call */
{
  register unsigned char *CutPt;

  LeftLimit = lm;          /* give value to C's global variable */
  EndOfText = eot;        /* give value to C's global variable */
  RightMargin = rm;       /* give value to C's global variable */

  /* Back searching to find cut point until left limit */

  if (isbpunc(*(RightMargin + 1)) || istspmk(*(RightMargin + 1)))
    --RightMargin;
  for (Indx = RightMargin; Indx >= LeftLimit; --Indx)
    {
      if ((CutPt = ThaiEngCutRtn(Indx)) != NONE)
        return((CutPt >= LeftLimit)? CutPt : NONE);
      else
        if ((CutPt = (*FuncPtr[*Indx])(Indx)) != NONE)
          if (CutPt >= LeftLimit && CutPt <= RightMargin
              && !isfpunc(*CutPt) && !isbpunc(*(CutPt+1))
              && !istspmk(*(CutPt+1)))
            return (CutPt);
    }
  //return(RightMargin); /* Reach LeftLimit, return RightMargin */
  return(LeftLimit-1);
}

int getdic( char *fname )
{
  FILE *fp;
  char str[KEYWORD_MAX];
  unsigned i;

  if( (fp=fopen( fname ,"r")) == NULL ) {
    printf("Error: cannot find %s",fname );
    exit(1);
  }

  fscanf(fp,"%s",str);
  keyword_key=strlen(str);
  while(!feof(fp)) {
    if((i=strlen(str))>1) {
      strcpy(dic[dic_key].word,str);
      dic_key++;
      if(keyword_key<i) keyword_key=i;
    }
    fscanf(fp,"%s",str);
  }
  fclose(fp);

  if( verbose ) {
    printf("Max length of keyword : %d\n",keyword_key);
    printf("Number of keyword : %d\n",dic_key);
  }
  return 0;
}

```

```

int indic(char *str) {
    int low,high,mid;

    low=0;
    high=dic_key-1;
    while(low<=high) {
        mid=(low+high)/2;
        if(strcmp(str,dic[mid].word)<0)
            high=mid-1;
        else
            if(strcmp(str,dic[mid].word)>0)
                low=mid+1;
            else
                return 1; //mid;
    }
    return NONE;
}

#ifdef _THAI_MAIN_

int _isspace(unsigned char ch );
int _isinspace( char * );

#define MAX_CHAR_LEN 500

int main(int argc , char *argv[] ) {
    char result[ MAX_CHAR_LEN ],tmpStr[ MAX_CHAR_LEN ], words[ MAX_CHAR_LEN ];
    char buffer[ MAX_CHAR_LEN ];
    int nPos,nArg,nResult,n,i,nLen;
    int inword , indict, nCondition;

    if( argc == 1 ) {
        printf("Usage: tsepword -w word1 word2 ... -d [path\\]dictionary \n");
        exit(0);
    }

    verbose = 0; // no echo initial value
    words[0] = '\0';

    inword = indict = 0;
    nArg = 1;

    while( nArg < argc ) {
        if( argv[nArg][0] == '-' || (!inword && !indict) ) {
            if( argv[nArg][0] != '-' ) {
                nArg++;
                continue;
            }
            inword = argv[nArg][1] == 'w';
            indict = argv[nArg][1] == 'd';

            nArg++;
            continue;
        }

        if( indict ) {
            dictName = (char *) malloc( sizeof(char)* (strlen(argv[nArg])+1) );
            strcpy( dictName , argv[nArg++] );
            indict = 0;

```

```

        initialize( dictName );
        continue;
    }

    if( inword ) {
        sprintf( words , "%s%s%s" , words , (words[0] != '\0' ? " " : "" ) , argv[nArg] );
    }

    nArg++;
}

tmpStr[0] = '\0';
result[0] = '\0';

n = i = 0;
while( 1 ) {

    if( words[i] && words[i] != ' ' ) {
        tmpStr[n++] = words[i++];
        continue;
    }

    tmpStr[n] = '\0';

    if( xlen(tmpStr) != 0 ) {
        thaiword( buffer , tmpStr );
        if( xlen( buffer ) != xlen(tmpStr) ) {
            sprintf( result , "%s%s(%s)" , result , (result[0] != '\0' ? " " : "" ) , buffer );
        } else {
            sprintf( result , "%s%s%s%s" , result ,
                (result[0] != '\0' ? " " : "" ) ,
                tmpStr ,
                tmpStr[0] >= 'ก' ? " " : "" );
        }
        n = 0;
    }

    while( words[i] && words[i] == ' ' )
        i++;

    if( !words[i] )
        break;
}

printf( "\n%s\n" , result );
return 0;
}

int lstrcmp(char *s , char *t) {
    int i = 0;
    while( s[i] && t[i] && (char) tolower(s[i]) == (char) tolower(t[i]) ) {
        i++;
    }
    if( !s[i] && !t[i] )
        return 0;

    return s[i] ? 1 : -1;
}

```

```

int xlen(char *s) {
    int t,e;

    if( s[0] == '\0' )
        return 0;
    t = 0;
    while( s[t] && s[t] == ' ' ) t++; // skip blank
    e = strlen(s)-1;
    while( s[e] && s[e] == ' ' ) e--; // skip blank
    return (e-t)+1;
}

#endif _THAI_MAIN_

int _isspace(unsigned char ch) {
    return ch == ' ' || ch == 9 || ch == 160;
}

int _isinspace(char *str) {
    int i=0;
    while( str[i] && _isspace( str[i] ) ) // skip space first
        i++;

    while( str[i] && !_isspace( str[i] ) ) // find space in str
        i++;

    if( !str[i] ) // end with no space
        return 0;

    while( str[i] && _isspace( str[i] ) ) // skip space last
        i++;

    if( !str[i] ) // end with space
        return 0;

    return 1; // there is space in word
}

```

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

1.2.10 makefile      ข้อกำหนดเกี่ยวกับการแปลโปรแกรม

# for gmain.exe

CC= gcc

CFLAGS= -O2

OBJS= gmain.o gnfile.o gnstring.o gnmem.o gread.o \  
 thaiword.o thaiutl1.o thaiutl2.o thaiutl3.o thaiutl4.o \  
 thaiutl5.o

.c.o:

\$(CC) -c \$(CFLAGS) \$<

all: gmain

gmain: \$(OBJS)  
 \$(CC) -o gmain \$(CFLAGS) \$(OBJS)

\$(OBJS): makefile gnconfig.h

1.3.1 tsepword.mak      ข้อกำหนดเกี่ยวกับการแปลโปรแกรม

# for tsepword.exe

CC= gcc

CFLAGS= -O2 -D\_THAI\_MAIN\_

OBJS= thaiword.o thaiutl1.o thaiutl2.o thaiutl3.o thaiutl4.o thaiutl5.o

.c.o:

\$(CC) -c \$(CFLAGS) \$<

all: tsepword

tsepword: \$(OBJS)  
 \$(CC) -o tsepword \$(CFLAGS) \$(OBJS)

\$(OBJS): tsepword.mak

สถาบันวิทยบริการ  
 จุฬาลงกรณ์มหาวิทยาลัย



ส่วนติดต่อกับผู้ใช้

## 2.1 admin.htm เว็บเพจหลักสำหรับจัดการระบบ

```
<html>

<head><title>ระบบจัดเตรียมแฟ้มข้อมูลสำหรับสนับสนุนการค้นคืนบทคัดย่อ</title></head>
<frameset framespacing="0" border="false" frameborder="0" rows="23%,*">
  <frame name="Admin header" scrolling="no" noresize src="admin_h.htm">
  <frame name="Buttom" scrolling="auto" src="/ts-bin/admin.pl?lo=load&htm=login">
  <noframes>
    <body>
      <p>This page uses frames, but your browser doesn't support them.</p>
    </body>
  </noframes>
</frameset>
</html>
```

## 2.2 admin\_h.htm เว็บเพจส่วนหัวสำหรับ admin.htm

```
<html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-874">
<meta http-equiv="pragma" content="no-cache">
</head>

<body bgcolor="#FFFFFF">
<basefont face="AngsanaUPC, BrowalliaUPC, CordiaUPC, DilleniaUPC, EucrosiaUPC" size="3">

<table border="0" width="100%" bgcolor="#800000" cellspacing="0" cellpadding="0">
  <tr>
    <td width="100%"><p align="center"></td>
  </tr>
</table>
<p align="center">&nbsp;</p>
</body>
</html>
```

## 2.3 index.htm เว็บเพจแสดงหัวข้อเลือกเริ่มต้น

```
<html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-874">
<meta http-equiv="pragma" content="no-cache">
<title>ระบบสนับสนุนการค้นคืนบทคัดย่อวิทยานิพนธ์</title></head>

<body bgcolor="#FFFFFF">
<basefont face="AngsanaUPC, BrowalliaUPC, CordiaUPC, DilleniaUPC, EucrosiaUPC" size="3">

<table border="0" width="100%" bgcolor="#800000" cellspacing="0" cellpadding="0">
  <tr>
    <td width="100%"><p align="center"></td>
  </tr>
```

```

</table>

<p align="center">&nbsp;</p>
<div align="center"><center>

<table border="0" cellpadding="0" cellspacing="0" width="28%">
  <tr>
    <td width="100%"><p align="center"><a href="search.htm">
      </a>
    </td>
  </tr>
  <tr>
    <td width="100%"><p align="center"><a href="admin.htm">
      </a></td>
    </tr>
</table>
</center></div>
</body>
</html>

```

#### 2.4 search.htm เว็บไซต์หลักสำหรับการระบบข้อคำถามเพื่อค้นคืน

```

<html>

<head>
<title>ค้นหาข้อมูลบทคัดย่อวิทยานิพนธ์-จุฬาลงกรณ์มหาวิทยาลัย</title>
</head>

<frameset framespacing="0" border="false" frameborder="0" rows="41%,*">
  <frame name="Search" scrolling="no" noresize src="s_main.htm">
  <frame name="Target" src="s_help.htm" scrolling="auto">
  <noframes>
    <body>
      <p>This page uses frames, but your browser doesn't support them.</p>
    </body>
  </noframes>
</frameset>
</html>

```

#### 2.5 s\_help.htm เว็บไซต์แสดงรายละเอียดช่วยเหลือสำหรับข้อคำถามเพื่อค้นคืน

```

<html>

<head>
<title>อธิบายวิธีค้นหา</title>
</head>

<body bgcolor="#FFFFFF">
<div align="center"><center>
<table border="1" width="62%" cellpadding="0" cellspacing="0">
  <tr>
    <th width="100%" align="center" bgcolor="#808080" colspan="2"><font
      face="AngsanaUPC,CordiaUPC" color="#FFFFFF"><big>
      วิธีการค้นหาเพื่อให้ได้ข้อมูลที่ตรงต่อความต้องการ</big></font></th></tr>
  <tr>
    <th width="36%" align="center" bgcolor="#808080"><font face="AngsanaUPC,CordiaUPC"

```

```

color="#FFFFFF"><strong>ข้อความ</strong></font></th>
<th width="64%" bgcolor="#808080"><p align="center"><font face="AngsanaUPC,CordiaUPC"
color="#FFFFFF"><strong>ความหมาย</strong></font></th>
</tr>
<tr>
<td width="36%" align="center" bgcolor="#FFFFFF"><font color="#000000">w1</font></td>
<td width="64%" bgcolor="#FFFFFF"><font color="#000000">&nbsp;<font
face="AngsanaUPC,CordiaUPC">หาเอกสารที่มีข้อความ w1 อยู่ </font></td>
</tr>
<tr>
<td width="36%" align="center" bgcolor="#C0C0C0"><font color="#000000">w1 w2</font></td>
<td width="64%" rowspan="4" bgcolor="#C0C0C0">
<font color="#000000" face="AngsanaUPC,CordiaUPC">&nbsp;<
หาเอกสารที่มีข้อความทั้ง w1 และ w2 อยู่</font></td>
</tr>
<tr>
<td width="36%" align="center" bgcolor="#C0C0C0">
<font color="#000000">&quot;w1 w2&quot;</font></td>
</tr>
<tr>
<td width="36%" align="center" bgcolor="#C0C0C0"><font color="#000000">w1+w2</font></td>
</tr>
<tr>
<td width="36%" align="center" bgcolor="#C0C0C0"><font color="#000000">w1 and w2</font></td>
</tr>
<tr>
<td width="36%" align="center"><font color="#000000">w1 or&nbsp;<font color="#000000" face="AngsanaUPC,CordiaUPC">&nbsp;<
หาเอกสารที่มีข้อความ w1 หรือข้อความ w2</font></td>
</tr>
<tr>
<td width="36%" align="center" bgcolor="#C0C0C0"><font color="#000000">w1 not w2</font></td>
<td width="64%" bgcolor="#C0C0C0"><font color="#000000"
face="AngsanaUPC,CordiaUPC">&nbsp;<หาเอกสารที่มีข้อความ w1 แต่ไม่มีข้อความ w2 </font></td>
</tr>
<tr>
<td width="36%" align="center" bgcolor="#FFFFFF"><font color="#000000">w*</font></td>
<td width="64%" bgcolor="#FFFFFF">
<font color="#000000" face="AngsanaUPC,CordiaUPC">&nbsp;<ค้นหาเอกสารที่มีค่าขึ้นต้น w</font>
</td>
</tr>
</table>
</center></div>
<p align="center"><font face="AngsanaUPC,CordiaUPC">
เพื่อความสมบูรณ์ของการแสดงผลควรปรับขนาดจอ 800x600 pixels</font></p>
</body>
</html>

```

## 2.6 s\_main.htm เว็บไซต์สำหรับระบุข้อความเพื่อดำเนิน

```

<html>

<head>
<title>ระบบสนับสนุนการค้นคืนบทคัดย่อวิทยานิพนธ์ผ่านเครือข่าย</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-874">
<meta http-equiv="pragma" content="no-cache">
</head>

<body bgcolor="#FFFFFF">

<form NAME="Search" METHOD="GET" ACTION="/ts-bin/search.pl" align="center" target="Target">

<table border="0" width="100%" bgcolor="#800000" cellspacing="0" cellpadding="0">
<tr>
<td width="100%" colspan="3"><div align="center"><center><p> 
</td>
</tr>
<tr align="center">
<td width="36%">&nbsp;</td>
<td width="32%"><div align="center"><center><p><strong> <font face="AngsanaUPC,
BrowalliaUPC, CordiaUPC, DilleniaUPC" olor="#FFFFFF">
<big>ระบุข้อความที่ต้องการค้นหา </big></font></strong>
</td>
<td width="32%" align="center"><div align="left"><p>
<a href="s_help.htm" target="Target">
<font face="AngsanaUPC, BrowalliaUPC, CordiaUPC, DilleniaUPC"
color="#FFFFFF" size="2"><i>วิธีการค้นหา</i></font></a>
</td>
</tr>
<tr align="center">
<td width="100%" colspan="3"><div align="center"><center><p>
<font face="Ms Sans Serif,AngsanaUPC, BrowalliaUPC, CordiaUPC,
DilleniaUPC">
<input type="text" name="sf" size="41">
<input type="submit" name="Submit" value="ค้นหา"></font>
<input type="reset" value=" ลบ " name="ClearBtn">
</td>
</tr>
</table>
<basefont face="AngsanaUPC, BrowalliaUPC, CordiaUPC, DilleniaUPC, EucrosiaUPC" size="3"><div
align="center"><center>
<table border="0" width="100%" bgcolor="#C0C0C0" cellspacing="0" cellpadding="0">
<tr>
<td width="17%" valign="top" align="center" bgcolor="#800000">
<div align="center"><center><p>
<font face="AngsanaUPC, BrowalliaUPC, CordiaUPC, DilleniaUPC,
EucrosiaUPC" color="#FFFFFF">ค้นหาเฉพาะ</font>
</td>
<td width="14%" valign="top" align="center" bgcolor="#808080">
<div align="center"><center><p>
<font color="#FFFFFF"><big>
<input type="checkbox" name="tp" value="1"></big>
<font face="AngsanaUPC, BrowalliaUPC, CordiaUPC, DilleniaUPC,

```



```

</form>
</body>
</html>

```

2.8 menu.htm เว็บไซต์ที่เรียกโดย admin.pl เพื่อแสดงหัวข้อการทำงาน

```

<html>

<head>
<title>ปรับปรุงเพิ่ม</title>
<base target="main">
</head>

<body><div align="center"><center>
<table border="0" cellpadding="0" cellspacing="0" width="121">
<tr>
<td width="121" align="center" bgcolor="#FFFFFF"><p align="center">
<a href="/ts-bin/admin.pl?%LINK%&lo=load&htm=getfname" target="board">

</a></td>
</tr>
<tr>
<td width="121" align="center" bgcolor="#FFFFFF">
<a href="/ts-bin/admin.pl?%LINK%&lo=load&htm=gettext" target="board">

</a></td>
</tr>
<tr>
<td width="121" align="center" bgcolor="#FFFFFF">
<a href="/ts-bin/admin.pl?%LINK%&lo=logout" target="board">

</a></td>
</tr>
</table>
</center></div>
</body>
</html>

```

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## โปรแกรมบทคำสั่งซีไอ

## 3.1 search.pl สำหรับประมวลผลข้อความคำถามสำหรับค้นคืน

```

# search.pl
# cgi from s_main.htm (search.htm)
# form field variable:
# 1. sf  textbox  string for searching
# 2. ad  checkbox '1' = search in meta ADVISOR
# 3. tp  checkbox '1' = search in meta TOPIC
# 4. kw  checkbox '1' = search in meta KEYWORD
# 5. au  checkbox '1' = search in meta AUTHOR
# 6. np  dropdown total result for each page
# 7. pg  hidden   pageno
#

require "c:\\thesis\\root\\ts-bin\\mylib.pll";

*hVarReq = ();
readVar( \%hVarReq );

#declare Global variable
*baseCGI  = ();
*baseDOC  = ();

$baseDir = &baseDIR;
$baseCGI = ($baseDir . "ts-bin\\");
$baseDOC = ($baseDir . "ts-docs\\");

$qry = $ENV{ "QUERY_STRING" };
($qry,$a1) = split( /&pg=/, $qry , 2 );
$qry = "search.pl?" . $qry . "&pg=";

#Inspect another user currently use

$chkFile = ($baseCGI."doing.log");
if( -e $chkFile ) {
    print "Content-type: text/html\n\n";
    print "<body bgcolor=\"#FFFFFF\"><center>\n";
    print "<p>Indexed file being process by another request<br>\n";
    print "</body>\n";
    exit 0;
}

$indexfile = ($baseCGI . "thesis.idx");
$tsepword  = ($baseCGI . "tsepword.exe");
$thaidict  = ($baseCGI . "Newdict.txt");
$swish     = ($baseCGI . "swish.exe");
$swcfg     = ($baseCGI . "swish.cfg");

htmError( "Indexed file not available" ) if !( -e $indexfile );
htmError( "tsepword.exe not found" )   if !( -e $tsepword );
htmError( "swish.exe not found" )     if !( -e $swish );

$CText = $hVarReq{ "sf" };
htmError( "specified message for searched" ) if !( $CText );

```

```

$cSearch = "";
open( FP , "$tsepword -d $thaidict -w $cText |" );
foreach $line (<FP>) {
    $cSearch .= $line;
}

close( FP );

# protect '*'
$_ = $cSearch;
while(/\/\*/) {
    $cSearch =~ s/\/\*/^*/g;
    $_ = $cSearch;
}

$line = "";
$line .= " TOPIC=" . $cSearch if $hVarReq{ "tp" };
$line .= "or AUTHOR=" . $cSearch if $hVarReq{ "au" };
$line .= "or ADVISOR=" . $cSearch if $hVarReq{ "ad" };
$line .= "or KEYWORD=" . $cSearch if $hVarReq{ "kw" };
$line = $cSearch if !$line;

open( FP , "$swish -w $line -f $indexfile -C $swcfg |" );

@result = ();
$nFound = 0;
$pagePl = 0+$hVarReq{ "np" }; # line per page default = 10
$pageNo = 0+$hVarReq{ "pg" }; # page number default = 0
$pageSt = ($pageNo*$pagePl)+1; # default 1
$searchErr = "";

while( <FP> ) {
    if( /\/\#/ ) {
        next;
    }
    if( /\/^err/ ) {
        if( /common/ ) {
            $searchErr = "คำที่ให้ค้นหาเป็นคำสามัญเกินไป ";
        } else {
            $searchErr = "ไม่พบข้อความที่ต้องการค้นหา ";
        }
    }
    break;
} else {

    $line = $_;
    if( length($line) > 2 ) {
        $nFound++;
        if( $nFound >= $pageSt && $nFound < ($pageSt+$pagePl) ) {
            ($rank,$link,$line) = split( // , $line , 3 );
            $line = substr( $line , 1);
            ($topic,$rate) = split( // , $line , 2);
            $link =~ s/SERVER//;
            ($a1,$a2) = split( // , $link , 2);
            ($a2,$a1) = split( \./ , $a2 , 2);
            $link = "/ts-docs/absdocs/" . $link;

            $fg = $nFound%2 ? "#0000FF" : "#FF0080";
            $bg = $nFound%2 ? "#FFFFFF" : "#C0C0C0";

```





```

print "<br>\n";
if( length($searchErr) > 0 ) {
    print "<p> <font face=\\"AngsanaUPC,CordiaUPC\\">$searchErr</font><br>\n";
}
print "<center>\n";
print "<table border=\\"1\" cellspacing=\\"1\" width=\\"96%\">\n";
foreach $I (@result) {
    print "<tr>\n";
    print "<font face=\\"AngsanaUPC,CordiaUPC\\">$I</font>\n";
    print "</tr>\n";
}
print "</table></center>\n";
print "</body>\n";
print "</html>\n\n";

exit 0;
#----- END OF MAIN -----

```

### 3.2 admin.pl สำหรับประมวลผลการจัดเตรียมเพิ่มข้อมูลแบบระยะไกล

```

# Admin
# 1. lo call module (load)

require "c:\\thesis\\root\\ts-bin\\mylib.pll";

*hVarReq = ();
readVar( \%hVarReq );

#declare Global variable
*baseCGI = ();
*baseDOC = ();
$timeOut = (60 * 60 * 3); # 3 hrs

$baseDir = &baseDIR;
$baseCGI = ($baseDir . "ts-bin\\");
$baseDOC = ($baseDir . "ts-docs\\");

# check called module
&htmlError( "Error: called module not found , Please contact admin" ) if $hVarReq{ "lo" } eq "";

if( -e ($baseCGI . "doing.txt") ) {
    if( !($hVarReq{ "lo" } eq "logout") ) {
        &busy();
    }
}

if( $hVarReq{ "lo" } eq "load" ) {
    if( !($hVarReq{ "htm" } eq "login") ) {
        &chkid();
    }
    &loadhtm();
}

```

```

if( $hVarReq{ "lo" } eq "chklogin" ) {
    &chklogin();
    $hVarReq{ "htm" } = "manage";
    &loadhtm();
}

&chkid();

if( $hVarReq{ "lo" } eq "edit" ) {
    &editor;
}

if( $hVarReq{ "lo" } eq "save" ) {
    &saveEditor;
}

if( $hVarReq{ "lo" } eq "logout" ) {
    &logout;
}

if( $hVarReq{ "lo" } eq "reindex" ) {
    &reindex;
}

# not found function call

&htmlError( "Error: called module [" . $hVarReq{ "lo" } . "] not load , Please contact admin" );
exit 0;

# ----- END OF MAIN -----

# ===== START SUB ROUTINE =====

# ----- LOADHTM -----

sub loadhtm {
local $fn,$hide,$link;

    &htmlError( "Specified html file" ) if $hVarReq{ "htm" } eq "";
    $hide = "";
    $link = "&";
    if( $hVarReq{ "un" } ) {
        $hide = "<input type=\"hidden\" name=\"un\" value=\"\" . $hVarReq{ "un" } . "\">\n";
        $hide .= "<input type=\"hidden\" name=\"ui\" value=\"\" . $hVarReq{ "ui" } . "\">\n";
        $link = "un=" . $hVarReq{ "un" } . "&ui=" . $hVarReq{ "ui" } . "&";
    }

# text file
if( !( $hVarReq{ "tx" } eq "" ) ) {
    $link .= "tx=" . $hVarReq{ "tx" } . "&";
}

$fn = $baseCGI . "admin\\" . $hVarReq{ "htm" } . ".htm";
&htmlError( "file [" . $fn . "] not found" ) if !( -e $fn );
open( FP , $fn );
while( <FP> ) {
    if( /%ID%/ ) {
        s/%ID%/$hide/;
    }
    if( /%LINK%&/ ) {

```

```

        s/%LINK%&/$link/;
    }
    print $_;
}
close( FP );
exit 0;
}

```

```
# ----- EDITOR -----
```

```

sub editor {
local $fn,@al,$data , $ht ,$hide ,$save;

    &htmlError( "ระบุชื่อเพิ่ม " ) if $hVarReq{ "fn" } eq "" ;
    $_ = $fn = $hVarReq{ "fn" };
    if( /\V ) {
        @al = split( /\V , $fn );
        $fn = @al[ $#al ];
        # last index value
    }

    $_ = $fn = $baseCGI . lc( $fn );

    if( !( /.htm/ || /.lst/ ) ) {
        &htmlError( "นามสกุลเพิ่มต้องเป็น '.htm' หรือ '.lst' " );
    }

    $data = "";
    $save = "เพิ่ม";
    if( -e $fn ) {
        open( FP , $fn );
        while( <FP> ) {
            $data .= $_;
        }
        close( FP );
        $save = "แก้ไข";
    }

    $hide = "<input type=\"hidden\" name=\"un\" value=\"\" . $hVarReq{ "un" } . "\">\n";
    $hide .= "<input type=\"hidden\" name=\"ui\" value=\"\" . $hVarReq{ "ui" } . "\">\n";

    $ht = $baseCGI . "admin\editor.htm";
    &htmlError( "file [ " . $ht . " ] not found" ) if !( -e $ht );
    open( FP , $ht );
    while( <FP> ) {
        if( /%ID%/ ) {
            s/%ID%/$hide/;
        }
        if( /%FILE_NAME%/ ) {
            s/%FILE_NAME%/$fn/;
        }
        if( /%SAVE%/ ) {
            s/%SAVE%/$save/;
        }
        if( /%DATA%/ ) {
            s/%DATA%/$data/;
        }
    }
}

```

```

    }
    print $_;
}
close( FP );
exit 0;
}

# ----- SAVE EDITOR -----

sub saveEditor {
local $fn,@al,$data , $ht , $hide , $save;
    $fn = $hVarReq{ "fn" };
    open( FP , ">$fn" );
    $hVarReq{ "dt" } =~ s/\
//g;
    print FP $hVarReq{ "dt" };
    close( FP );
    &htmlError( "ข้อมูลถูกบันทึกลงแฟ้ม $fn เรียบร้อย " );
}

# ----- CHKLOGIN -----

sub chklogin {
local $fname,$un,$pwd;

    srand( time | $$ ); # seed random

    $fname = $baseCGI . "user.db";
    open( FP , $fname ) or htmlError( "Cannot open user file" );
    $ui = -1;
    while( <FP> ) {
        chop;
        ($un,$pwd) = split(// , $_);
        if( $un eq $hVarReq{ "n" } && $pwd eq $hVarReq{ "p" } ) {
            $hVarReq{ "ui" } = int( rand() * 100 );
            $hVarReq{ "un" } = $hVarReq{ "n" };
            $fname = $baseCGI . "session\\" . $un . "." . $hVarReq{ "ui" };
            close(FP);
            open( FP , ">$fname" ) or htmlError( "session fail" );
            print FP time . "\n";
            print FP gmtime(time) . "\n";

            close( FP );
            return;
        }
    }
    close( FP );
    htmlError( "Invalid user name or password" ) if $ui < 0;
}

```

```

# ----- CHKID -----

sub chkid {
local $fn , $stm;
  if( $hVarReq{ "un" } eq "" || $hVarReq{ "ui" } eq "" ) {
    htmError( "This page forbidden" );
  }
  $fn = $baseCGI . "session\\" . $hVarReq{ "un" } . "." . $hVarReq{ "ui" };
  if( !( -e $fn ) ) {
    htmError( "Unauthorized prohibited or invalid session" );
  }
  open( FP , $fn );
  $stm = <FP>;
  close( FP );

  if( time > $stm + $timeOut ) {
    unlink( $fn );

    print <<EOF";
  pragma: no-cache
  content-type: text/html;

  <html>

  <head>
  <meta http-equiv="pragma" value="no-cache">
  <title>Forward page</title>
  <script language="JavaScript">
    function newwin() {
      top.location = "/ts-docs/"
    }
  </script>
  </head>

  <body>
  <center>
  <H2>Timeout</H2>
  Please <a href="JavaScript:newwin()">click hear</a> for new login
  </center>
  </body>
  </html>

  EOF
    exit 0;
  }

  open( FP , ">$fn" ) or htmError( "session fail" );
  print FP time . "\n";
  print FP gmtime(time) . "\n";
  close( FP );
}

```

```

# ----- LOGOUT -----

sub logout {
local $fn;
  $fn = $baseCGI . "session\\" . $hVarReq{ "un" } . "." . $hVarReq{ "ui" };
  unlink( $fn );

print <<'EOF';
pragma: no-cache
content-type: text/html;

<html>

<head>
<meta http-equiv="pragma" value="no-cache">
<title>Forward page</title>
<script language="JavaScript">
  function newwin() {
    top.location = "/ts-docs/"
  }
</script>
</head>

<body onLoad="newwin()">
<center>

<H2>Sign out from system , Now I bring to main page</H2>
if no change for long time please <a href="JavaScript:newwin()">click hear</a>
</center>
</body>
</html>

EOF
exit 0;
}

# ----- BUSY -----
sub busy {

print <<'EOF';
pragma: no-cache
content-type: text/html;

<html>

<head>
<meta http-equiv="pragma" value="no-cache">
<title>Forward page</title>
<script language="JavaScript">
  function newwin() {
    location = "/ts-docs/admin/a_busy.htm"
  }
</script>
</head>

<body onLoad="newwin()">
<center>
<H2>Be indexing..</H2>

```

```

if no change for long time please <a href="JavaScript:newwin()">click hear</a>
</center>
</body>
</html>

```

```

EOF
exit 0;
}

```

```
# ----- REINDEX -----
```

```
# 1. get text file -> submit
#
#

```

```

sub reindex {
local $tx, $do, $txBook, $log, $template, $gnmain, $swish, $cmd;
local $swishcfg, $field, $dict, $idx, $idxtmp, $doctemp;
  if( $hVarReq{ "tx" } eq "" ) {
    &indexError( "Not specified text file" );
  }
  $tx = $baseCGI . "text\\" . $hVarReq{ "tx" };
  $_ = $tx;
  if( !/^./ ) {
    $tx .= ".txt";
  }

  if( !(-e $tx) ) {
    &indexError( "Not found " . $tx );
  }
  $indexBook = $baseCGI . "textbook.db";
  if( -e $indexBook ) {
    open( FP, $indexBook ) or indexError( "Cannot open textbook.db" );
    $tx = lc( $tx );
    while( <FP> ) {
      chop;
      $_ = lc( $_ );
      if( $_ eq $tx ) {
        close( FP );
        indexError( $tx . " was indexed " );
      }
    }
    close( FP );
  }
}

```

```

$do = $baseCGI . "doing.txt";
open( DO_FP, ">$do" );

```

```

$gnmain = $baseCGI . "gnmain.exe";
$swish = $baseCGI . "swish.exe";
$swishcfg = $baseCGI . "swish.cfg";
$field = $baseCGI . "field.lst";
$dict = $baseCGI . "newdict.txt";
if( !( -e $gnmain ) ||
  !( -e $swish ) ||
  !( -e $swishcfg ) ||
  !( -e $field ) ||
  !( -e $dict ) ) {
  close( DO_FP );
}

```



```

unlink( $do );
indexError( "Error: please check exist of files\n" .
    $gnmain . "\n" . $swish . "\n" .
    $swishcfg . "\n" . $field . "\n" .
    $dict );
}

$template = $baseCGI . "genhtm.htm";
if( !( -e $template ) ) {
    close( DO_FP );
    indexError( $template . " not found" );
}

# indexing
$log = $baseDOC . "status.txt";
open( LOG_FP , ">$log" );
print LOG_FP "เริ่มการจัดเตรียม " . gmtime(time+(7*60*60)) . "\n";
print LOG_FP "Start at " . gmtime(time+(7*60*60)) . "\n";
print LOG_FP "จัดเตรียมเพิ่มสำหรับแสดง ----- \n";

$cmd = "$gnmain -f $field -p $template -l $log -t $tx";

print LOG_FP $cmd . "\n";
close( LOG_FP );

system $cmd;

$cmd = $baseCGI . "deltemp.bat";
open( FP , "$cmd" );
close( FP );

system $cmd;

$template = $baseCGI . "genidx.htm";
if( !( -e $template ) ) {
    close( DO_FP );
    unlink( $do );
    indexError( $template . " not found" );
}

open( LOG_FP , ">>$log" );
print LOG_FP "จัดเตรียมเพิ่มสำหรับสร้างแฟ้มดัชนี ----- \n";

$cmd = "$gnmain -f $field -p $template -l $log -t $tx -T $dict";

print LOG_FP $cmd . "\n";
close( LOG_FP );

system $cmd;

# index by swish
$idix = $baseCGI . "thesis.idx";
$idixtmp = $baseCGI . "__zztmp.idx";
$doctemp = $baseCGI . "doctemp\\*. *";
$_ = $doctemp;
if( /:/ ) {
    ($a,$doctemp) = split( /:/ , $_ );
}

```

```

open( LOG_FP , ">>$log" );
print LOG_FP "สร้างแฟ้มดัชนี ----- \n";

if( -e $idx ) {
    $cmd = "$swish -i $doctemp -c $swishcfg -f $idxtmp -l $log";
    print LOG_FP $cmd . "\n";
    close( LOG_FP );
    system $cmd;

    # merge
    $cmd = "$swish -M $idxtmp $idx";
    system $cmd;

    unlink( $idxtmp );

} else {
    $cmd = "$swish -i $doctemp -c $swishcfg -f $idx -l $log";
    print LOG_FP $cmd . "\n";
    close( LOG_FP );
    system $cmd;
}

# done
open( LOG_FP , ">>$log" );
print LOG_FP "สิ้นสุดการจัดเตรียม " . gmtime(time+(7*60*60)) . "\n";
print LOG_FP "Finished at " . gmtime(time+(7*60*60)) . "\n";
close( LOG_FP );

close( DO_FP );
unlink( $do );

# save text name in index book
open( FP , ">>$indexBook" );
print FP $tx . "\n";
close( FP );

print <<'EOF';
pragma: no-cache
content-type: text/html;

<html>
<head>
<meta http-equiv="pragma" value="no-cache">
<title>Forward page</title>
<script language="JavaScript">
    function warn() {
        alert( "จัดเตรียมแฟ้มดัชนีเรียบร้อยแล้ว " );
        parent.location = "/ts-docs/admin/a_indexok.htm";
    }
</script>
</head>

```

```
<body onLoad="warn()">
```

```
</html>
```

```
EOF
  exit 0;
}
```

```
# ----- NO FOUND TEXT -----
```

```
sub indexError {
  local ($msg) = @_ if @_;
```

```
  $_ = $msg;
  if( /\V ) {
    $msg =~ s/\V\\g;
  }
```

```
  print <<'EOF';
  pragma: no-cache
  content-type: text/html;
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="pragma" value="no-cache">
```

```
<title>Forward page</title>
```

```
<script language="JavaScript">
```

```
  function warn() {
```

```
  EOF
```

```
  print "alert( \"$msg\" );\n";
```

```
  print <<'EOF';
    parent.location = "/ts-docs/blank.htm";
  }
```

```
</script>
```

```
</head>
```

```
<body onLoad="warn()">
```

```
</html>
```

```
EOF
  exit 0;
```

```
}
```

## 3.3 mylib.pll    เพิ่มห้องสมุด

```

# mylib.pll Perl Library
# -----
# all of sub , required by TS-BIN by Stk
# -----
# ----- START LIBRARY -----
# baseDIR() : return base of thesis root directory
# readVar() : extract value came from form page
#

sub baseDIR {
  if( $ENV{ 'TS-HOME' } ) {

    $baseDir = $ENV{ 'TS-HOME' };

  } else {

    $baseDir = "c:\\thesis\\root\\";
  }

  if( substr( $baseDir , length($baseDir) , 1) eq "\\" ) {
    $baseDir = substr( $baseDir , 1,length($baseDir)-1)
  }

  return $baseDir;
}

# -----
# This sub routine for extract variable called from Web Page
# -----

sub readVar {
  local (*in) = @_ if @_;
  local ($i, $key, $val);

  # Read in text
  if ( $ENV{'REQUEST_METHOD'} eq "GET" ) {

    $in = $ENV{'QUERY_STRING'};

  } elsif ( $ENV{'REQUEST_METHOD'} eq "POST" ) {

    $in = substr( <STDIN> , 0 , $ENV{'CONTENT_LENGTH'} );
  }

  @in = split(/&|/, $in);

```

```

foreach $i (0 .. $#in) {

    # Convert plus's to spaces
    $in[$i] =~ s/\+/ /g;

    # Split into key and value.
    ($key, $val) = split(/=,$in[$i],2); # splits on the first =.

    # Convert %XX from hex numbers to alphanumeric
    $key =~ s/%(..)/pack("c",hex($1))/ge;
    $val =~ s/%(..)/pack("c",hex($1))/ge;

    # Associate key and value
    $key =~ tr/A-Z/a-z/;
    $in{$key} .= "\0" if (defined($in{$key})); # \0 is the multiple separator
    $in{$key} .= $val;

}

return;
}

#
# Show error message to Browser
#
#
sub htmError {
    local ($msg) = @_ if @_;
    print "Content-type: text/html\n\n";
    print "<body bgcolor=#FFFFFF><center>\n";
    print "<font size=+2>\n";
    print "<B>" . $msg . "</B><br>\n";
    print "</font></body>\n";
    exit 0;
}

#
# Header for all returned message from CGI
#
sub prnHead {
    #-----print header-----
    print "Content-type: text/html\n\n";
    print "<!doctype HTML public \"-//W30 //DTD W3 HTML 3.0//EN\">\n";
    print "<html>\n\n<head>\n";
    print "<meta http-equiv=\"Content-Type\" content=\"text/html;\">\n";
    print "</head>\n";
    print "<body bgcolor=#FFFFFF>\n";
}

#
# Footer for all returned message from CGI
#
sub prnFoot {
    print "</body></html>\n";
    exit 0;
}

```

# ----- END LIBRARY -----

#this line must be not changed: 1; tell perl to execute EOF  
1;



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## ประวัติผู้เขียน

นายไสภณ ทองเกษม เกิดที่อำเภอดำเนินสะดวก จังหวัดราชบุรี สำเร็จการศึกษาปริญญาตรีบริหารธุรกิจ (การบัญชี) มหาวิทยาลัยรามคำแหง เข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต (นอกเวลาราชการ) สาขาวิทยาศาสตร์คอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ. 2539



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย