รายงานสหกิจศึกษาฉบับสมบูรณ์

เครื่องมือสำหรับการประมวลผลข้อมูล

Tool for Data Processing

นางสาวฐาปณีย์ บุญชอบ

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2562

ชื่อโครงงานสหกิจศึกษา เครื่องมือสำหรับการประมวลผลข้อมูล
ชื่อ-สกุล นักศึกษา นางสาวฐาปณีย์ บุญชอบ
คณะ วิศวกรรมศาสตร์ **ภาควิชา** วิศวกรรมคอมพิวเตอร์
ชื่อ-สกุล อาจารย์นิเทศ ผศ.บัณฑิต พัสยา
ชื่อ-สกุล ผู้นิเทศงาน นายปิยณัฐ หนูอุไร
ชื่อสถานประกอบการ บริษัท เอเชีย ออนไลน์ พอร์ทัลส์ (ประเทศไทย) จำกัด

บทคัดย่อ

รายงานสหกิจศึกษาฉบับสมบูรณ์เล่มนี้มีวัตถุประสงค์เพื่อศึกษาและพัฒนาโปรแกรมที่ใช้ในการทำงานของบริษัท เอเชีย ออนไลน์ พอร์ทัลส์ (ประเทศไทย) จำกัด ซึ่งเป็นบริษัทในเครือของ Omniscien Technologies ซึ่งโปรแกรมที่พัฒนาบางส่วนเป็นผลิตภัณฑ์ที่ต้องการขายให้ลูกค้าจริง อาทิเช่น โปรแกรมนับคำ ประโยค บรรทัด ย่อหน้าและจำนวนรูปภาพในไฟล์ บางส่วนเป็นโปรแกรมที่พัฒนาขึ้นเพื่อลดระยะเวลาการทำงานในขั้นตอนของการเตรียมข้อมูล อาทิเช่น โปรแกรมตรวจสอบจำนวนคู่ภาษา โปรแกรมตรวจสอบอายุของไฟล์และวันที่มีการปรับปรุงไฟล์ล่าสุด โปรแกรมทำความสะอาดข้อมูลก่อนนำไปประมวลผล โปรแกรมแยกข้อความในไฟล์โดยภาษา โปรแกรมตรวจสอบและลบข้อความภาษาจีนแบบดั้งเดิม (Traditional) ออกจากไฟล์ บางส่วนเป็นโปรแกรมที่พัฒนาขึ้นเพื่อให้การทำงานของทีมมีความสะดวกสบายมากขึ้น อาทิเช่น โปรแกรมตรวจสอบและหาชื่อไฟล์ที่ยังไม่ถูกประมวลผล และบางส่วนเป็นโปรแกรมที่พัฒนาขึ้นเพื่อคอยตรวจสอบการทำงานของระบบและแจ้งเตือนเมื่อมีบางอย่างผิดปกติ อาทิเช่น โปรแกรมตรวจสอบการตัดประโยคและการตัดคำของเครื่องเซิร์ฟเวอร์ที่ใช้ในการทำงานในระบบ

โปรแกรมทั้งหมดที่กล่าวมาข้างต้น พัฒนาโดยภาษาจาวาและเชลล์ สคริปต์เป็นส่วนใหญ่ โดยเน้นการทำงานบนระบบปฏิบัติการลินุกซ์ ซึ่งแต่ละโปรแกรมสามารถนำไปใช้กับข้อมูลจริงได้เป็นอย่างดี ช่วยลดระยะเวลาการทำงานของทีมได้ ซึ่งถือว่าเป็นเวลาที่น่าพอใจ เนื่องจากมีการทำงานที่ถูกต้องแม่นยำกว่ามนุษย์และลดระยะเวลาในการทำงานได้เป็นอย่างดี

คำสำคัญ : เชลล์ สคริปต์, ระบบปฏิบัติการลินุกซ์, ภาษาจาวา

Co-operative Title: Tool for Data Processing
Student Intern Name: Thapanee Boonchob
Faculty: Engineering **Department:** Computer Engineering
Advisor Name: Asst.Poof.Bundit Pasaya
Mentor Name: Piyanat Noourai
Company: Asia Online Portals (Thailand) Limited

ABSTRACT

This complete cooperative education report aims to study and develop the work programs of Asia Online Portal (Thailand) Co., Ltd., a subsidiary of Omniscien Technologies, whose programs are partially developed as products to be sold to real customers such as program to count words, sentences, lines, paragraphs and the number of images in the file. Some of them are programs developed to reduce the time required for the data preparation process, such as language pairing programs. The program checks the age of the file and the date that the file was last updated. Data cleaning program before processing, program to extract text in files by language, program to checks each line is traditional Chinese or simplified Chinese and removes traditional Chinese from the file. Some are programs that are developed to make team work more convenient, such as programs to check and search for filenames that have not been processed. And some are programs that are developed to monitor system operations and provide warnings if there are have some faults, such as program to check server that run sentence segment and tokenize processing in the system.

All programs mentioned above developed by Java and Shell script by focusing on working on the Linux operating system. Each program can be used with real data as well. Can reduce the work time of the team. Which is considered a satisfactory time because there is more accurate than humans do it by themselves and reduces the time to work very well.

Keywords: Java Programming Language, Linux Operating System, Shell Script

กิตติกรรมประกาศ

รายงานสหกิจศึกษาฉบับสมบูรณ์เล่มนี้ สำเร็จได้ด้วยความอนุเคราะห์และการสนับสนุนอย่างดียิ่งของ ผศ.บัณฑิต พัสยา ซึ่งเป็นอาจารย์ที่ปรึกษาและอาจารย์นิเทศที่ให้คำปรึกษาและเข้าร่วมการนิเทศการทำสหกิจศึกษาที่บริษัท ตลอดจนตรวจสอบ ชี้แนะข้อผิดพลาดและข้อควรแก้ไขของรายงานสหกิจศึกษาฉบับสมบูรณ์เล่มนี้ นอกจากนี้ต้องขอขอบคุณบริษัท เอเชีย ออนไลน์ พอร์ทัลส์ (ประเทศไทย) จำกัด ที่เปิดรับนักศึกษาสหกิจศึกษา ทำให้ได้ประสบหน้าที่การทำงานจริง และขอขอบคุณ Gregory Binger, Dion Wiggins ผู้ร่วมก่อตั้งบริษัท นายปิยณัฐ หนูอุไร ผู้จัดการโครงการของทีม และสมาชิกภายในทีมทุกคนที่คอยให้คำแนะนำและให้ความช่วยเหลือในการทำงานเป็นอย่างดี

ฐาปณีย์ บุญชอบ

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตารางที่ **หน้า**

สารบัญภาพ

ภาพที่ **หน้า**

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

บริษัท เอเชีย ออนไลน์ พอร์ทัลส์ (ประเทศไทย) จำกัด เป็นบริษัทในเครือ Omniscien Technologies เป็นผู้ผลิตชั้นนำระดับโลกในด้านการประมวลผลภาษาที่มีคุณภาพสูงและมีความปลอดภัยสูง การแปลภาษาด้วยเครื่อง (MT) และเทคโนโลยีและบริการการเรียนรู้ของเครื่องสำหรับการใช้งานเนื้อหาที่เข้มข้น โซลูชั่นที่หลากหลายของเราให้บริการลูกค้าจากอุตสาหกรรมต่างๆ รวมถึงอุตสาหกรรมรองรับหลายภาษา, บริการวิจัยออนไลน์, การเผยแพร่, E-Commerce, สื่อและความบันเทิง, การท่องเที่ยวออนไลน์, เทคโนโลยี, องค์กรและรัฐบาล

Omniscien Technologies ได้รับชื่อเสียงในด้านโซลูชั่นที่ทันสมัยด้วยแพลตฟอร์ม Language Studio และ E-Commerce Studio ขึ้นอยู่กับความต้องการของลูกค้า แพลตฟอร์มสามารถปรับใช้ในหลากหลายวิธีเพื่อรวมเข้ากับระบบประมวลผลข้อมูลภายในและการจัดการการแปลสำหรับอุตสาหกรรมการแปลและระบบอื่นๆ แพลตฟอร์มนำเสนอระดับการปรับแต่งและการควบคุมที่เหนือชั้นรวมถึงฟีเจอร์ก่อนและหลังการประมวลผลที่สมบูรณ์ช่วยให้ลูกค้าได้รับข้อมูลที่ซับซ้อนที่สุดเพื่อให้ได้ทั้งคุณภาพสูงและผลผลิตสูงเพื่อตอบสนองทุกกรณีการใช้งาน

บริษัทมีการทำงานครอบคลุมมากกว่า 550 คู่ภาษาทั่วโลกและด้วยโซลูชั่นเฉพาะอุตสาหกรรมจำนวนมาก Omniscien Technologies ยังคงเป็นคู่ค้าทางเลือกสำหรับลูกค้าที่มีความซับซ้อนในการประมวลผลข้อมูลปริมาณมากและการแปลด้วยเครื่อง

เนื่องจากบริษัท เอเชีย ออนไลน์ พอร์ทัลส์ (ประเทศไทย) จำกัด ก่อตั้งขึ้นเพื่อดำเนินธุรกิจที่เกี่ยวข้องกับการประมวลผลข้อมูลเพื่อการแปลภาษาเป็นหลัก ซึ่งรองรับหลากหลายภาษา และยังมีธุรกิจอื่นอีก อาทิเช่น การขายผลิตภัณฑ์สำหรับนำไปประมวลผลเอง การขายโปรแกรมต่างๆ เป็นต้น ซึ่งเมื่อมีการทำงานที่หลากหลาย แน่นอนว่าย่อมเกิดปัญหาที่ไม่คาดคิดขึ้นได้ นำไปสู่การทำโครงงานนี้ขึ้น เพื่อรับทราบปัญหาและหาหนทางแก้ไขอย่างรวดเร็ว เนื่องจากเวลาทุกวินาทีมีค่าสำหรับการธุรกิจ หากเกิดปัญหาขึ้นและมีการแก้ไขปัญหาที่ล่าช้า นั่นหมายความว่าบริษัทอาจเสียรายได้และเสียลูกค้าได้

นอกจากนี้แพลตฟอร์มที่บริษัทใช้ในการแปลภาษายังเป็นแพลตฟอร์มที่พัฒนาโดยนักพัฒนาภายในบริษัทเอง ซึ่งมีข้อดีคือสามารถแก้ไขและพัฒนาได้อย่างสม่ำเสมอ ซึ่งแน่นอนว่าเมื่อเทคโนโลยีพัฒนาขึ้น แพลตฟอร์มก็จะถูกพัฒนาขึ้นตลอดเวลา และเมื่อมีลูกค้ามากขึ้น ข้อมูลมมากขึ้น ทำให้บริษัทเจอเงื่อนไขหรือข้อยกเว้นด้านภาษาในการทำงานมากขึ้น เป็นอีกเหตุผลหนึ่งที่นำไปสู่การทำโครงงานนี้ขึ้นเพื่อหาจุดบกพร่อง หรือจุดที่สามารถปรับปรุงแก้ไขแพลตฟอร์ม เพื่อเพิ่มประสิทธิภาพการทำงานหรือลดระยะเวลาการทำงานลง

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 เพื่อศึกษาการทำงานของแพลตฟอร์มที่ใช้ในการประมวลผลข้อมูล

1.2.2 เพื่อพัฒนา Tools ต่างๆให้การทำงานของทีม Data Production สะดวกมากขึ้น

1.2.3 เพื่อลดระยะเวลาการทำงานและอำนวยความสะดวกให้แก่ทีมได้

1.3 ขอบเขตของการวิจัย

1.3.1 การวิจัยมุ่งเน้นศึกษาการทำงานของแพลตฟอร์มที่ใช้ในการประมวลผลข้อมูลที่มีชื่อเรียกว่า Dropfolder ภายในบริษัท เอเชีย ออนไลน์ พอร์ทัลส์ (ประเทศไทย) จำกัด

1.3.2 การวิจัยมุ่งเน้นการพัฒนา Tools เพื่ออำนวยความสะดวกและลดระยะเวลาในการทำงานของทีม

1.4 วิธีดำเนินการวิจัย

1.4.1 ศึกษาความต้องการและขอบเขตของโครงงาน

1.4.2 วิเคราะห์ความต้องการและวางแผนตารางการพัฒนาโครงงานให้สอดคล้องกับงานเพื่อให้โครงงานสำเร็จภายในระยะเวลาที่กำหนด

1.4.3 เลือกเทคโนโลยีสำหรับพัฒนาโครงงานให้เหมาะสมกับงาน

1.4.4 ศึกษาเทคโนโลยีที่เลือกใช้สำหรับการพัฒนาโครงงาน

1.4.5 ศึกษาและทำความเข้าใจเกี่ยวกับกระบวนการทำงานของเฟรมเวิร์คที่บริษัทใช้ในการประมวลผลข้อมูล

1.4.6 วิเคราะห์กระบวนการทำงานของเฟรมเวิร์คเพื่อหา Gap Analysis

1.4.7 พัฒนาโปรแกรมที่สามารถช่วยลดข้อผิดพลาดของกระบวนการทำงานของเฟรมเวิร์คหรือช่วยลดระยะเวลาในการทำงาน

1.4.8 ทดสอบและปรับปรุงการทำงานของโปรแกรมให้มีขั้นตอนการทำงานที่ถูกต้อง เหมาะสมและมีประสิทธิภาพมากขึ้น

1.4.9 สรุปผลและจัดทำเอกสารอธิบายการทำงานและโครงสร้างของโครงงาน

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ทราบและเข้าใจการทำงานของแพลตฟอร์ม Drop Folder

1.5.2 สามารถพัฒนา Tools ต่างๆที่สามารถนำมาใช้ในทีมได้

1.5.3 สามารถลดระยะเวลาการทำงานของทีมได้

บทที่ 2
แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ก่อนกล่าวถึงเฟรมเวิร์คและการทำงานของโปรแกรมต่างๆที่ผู้วิจัยได้พัฒนา ผู้วิจัยอยากกล่าวถึงแนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้องกับสิ่งที่พัฒนาก่อน เพื่อให้ผู้อ่านมีความเข้าใจพื้นฐานที่ตรงกัน

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 ระบบปฏิบัติการยูนิกซ์

ยูนิกซ์จัดอยู่ในกลุ่มระบบปฏิบัติการ (OS) แบบ Mutitasking หรือ Multiuser ซึ่งถือกำเนิดที่สถาบัน Bell Labs วัตถุประสงค์หลักที่พัฒนาขึ้นมาเพื่อเป็นแพลตฟอร์มสำหรับการเขียน Software เพื่อใช้รันในระบบอื่นๆ แต่ก็มีการขยายขอบเขตออกไปจนในที่สุดกลายเป็นระบบปฏิบัติการ ซึ่งลักษณะของ Unix คือใช้งานด้วยข้อความและเก็บข้อมูลเป็นลำดับชั้น มีเครื่องมือ Command ให้ใช้งานมากมาย และสามารถทำงานรวมกันโดยใช้ Pipe (|) เป็นตัวเชื่อม

ระบบของยูนิกซ์ถูกบริหารจัดการภายใต้โปรแกรมหลักคือ Kernel เพื่อใช้ในการเริ่ม/หยุดโปรแกรมอื่นๆ และใช้ในการจัดการ File System ในระดับล่าง อีกทั้งยังคอยจัดการ Resource ที่มีอยู่ให้ Program อื่นๆใช้งานได้โดยไม่ชนกัน

สำหรับ Unix จะมีโปรแกรมที่เรียกว่า เชลล์ยูนิกซ์ (Unix shell) สำหรับรับคำสั่งผ่าน Command Line ถ้าเทียบกับ Window ก็คือ cmd.exe โดยบนระบบปฏิบัติการยูนิกซ์ แต่ละผู้ใช้งานที่เข้ามาใช้งานสามารถเลือก Shell หลายแบบเพื่อใช้งานตามความต้องการที่ต่างกัน ดังนี้

- Login Shell ทำหน้าที่หลังจาก Login สำเร็จ ทั้งต่อเข้าเครื่องโดยตรง (Console) และต่อผ่าน Telnet
- Interactive Shell สามารถรับคำสั่ง แต่ไม่ได้เริ่มตั้งแต่ Login เช่น การเปิด Terminal
- Non-interactive Shell เป็นการทำงานเพื่อรันสคริปต์หรือชุดคำสั่ง

เชลล์ยูนิกซ์แบ่งออกเป็น 2 ประเภท ดังนี้

- Bourne Shell (sh)

เรียกชื่อตามผู้สร้าง “Stephen Bourne” โดยปกติแล้วชื่อโปรแกรมจะเป็น “sh” อยู่ที่ Path “/bin/sh” ซึ่งมีการพัฒนาต่อยอดออกมาอีกหลายชนิด เช่น Almquist Shell (ash), Bourne-Again Shell (bash), Debian Almquist Shell (dash), Korn Shell (ksh), MirBSD Korn Shell (mksh), Z Shell (zsh)

- C Shell (csh)

เขียนขึ้นโดย “Bill Joy” อาศัยพื้นฐานจากรูปแบบของ C โดยสามารถรองรับ Feature แบบ Interactive มากมาย ในหลายๆระบบปัจจุบันอาจจะมีการ link ไปที่ TENEX C Shell (tcsh) แทน ซึ่งมีการพัฒนาจาก csh ให้ดีขึ้น และ feature หลายอย่างก็ถูกนำไปใช้ใน shell ประเภทอื่นๆด้วย [1]

2.1.2 ระบบปฏิบัติการลินุกซ์

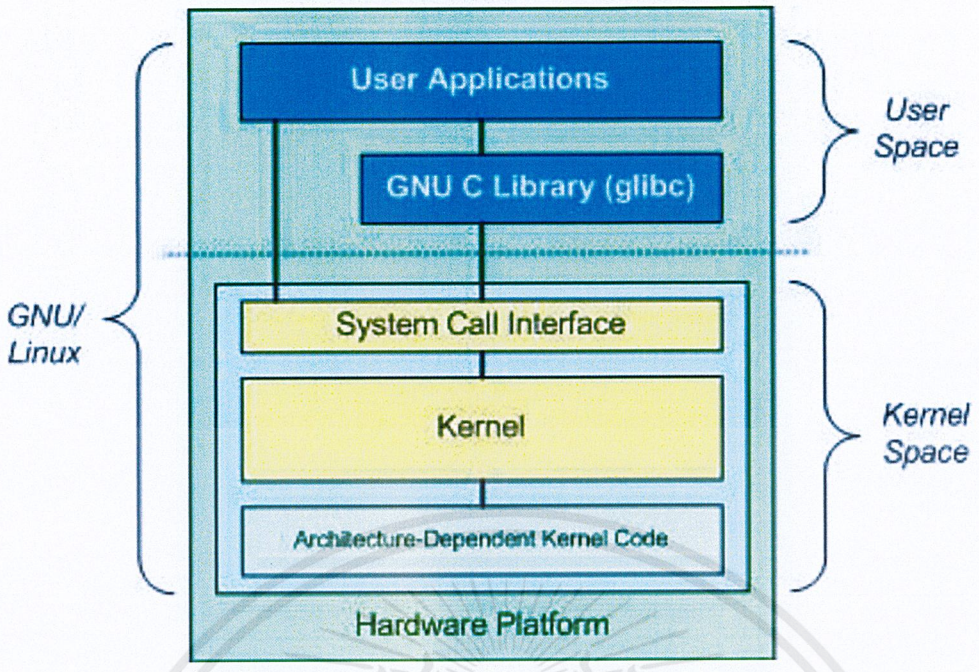
ลินุกซ์ คือระบบปฏิบัติการชนิดหนึ่งซึ่งเป็นระบบปฏิบัติการที่เป็น Open Source Software โดยมีการพัฒนาแจกจ่ายให้ผู้ใช้งานได้ฟรี ตามความหมายของ Linux แล้วจริงๆหมายถึง Linux Kernel หรือ Operating System Kernel ซึ่งทำหน้าที่เป็นตัวกลางเชื่อมต่อระหว่างฮาร์ดแวร์และระบบปฏิบัติการเพื่อบริหารจัดการทรัพยากรที่มีอยู่ให้เหมาะสม

เริ่มแรกระบบปฏิบัติการลินุกซ์เกิดขึ้นจากการพัฒนาบนคอมพิวเตอร์ที่ใช้ Chipset Intelx86 (32bit) แต่แล้วก็มีการพัฒนาให้รองรับกับแพลตฟอร์มอื่นทั่วไป เพราะการเข้าครอบครองตลาดของ Android บน Smartphone ทำให้ระบบปฏิบัติการลินุกซ์กลายเป็นระบบปฏิบัติการที่แพร่หลายมาก

ระบบปฏิบัติการลินุกซ์เริ่มต้นจากการคิดค้นระบบปฏิบัติการยูนิกซ์ขึ้นมาไว้สำหรับเป็นระบบปฏิบัติการสำหรับ Server และ ระบบขนาดใหญ่ ต่อมาได้มีคนพัฒนาระบบปฏิบัติการยูนิกซ์สำหรับ Personal Computer ขึ้นมาชื่อ “MINIX” ในปี 1987 แต่เนื่องจากอยู่ภายใต้ License สำหรับด้านการศึกษาเท่านั้น ทำให้นาย Linus Torvalds เริ่มพัฒนา Operating System Kernel ของตนเองขึ้นมาเพื่อใช้งาน ซึ่งสุดท้ายแล้วก็กลายมาเป็น Linux Kernel

ส่วนประกอบของ Linux Operation System มีดังนี้

- The Bootloader เป็นซอฟต์แวร์ที่ทำหน้าที่จัดการเรื่องการ Boot ของคอมพิวเตอร์ สำหรับผู้ใช้งานมันก็คือหน้าจอที่แสดงขึ้นมาช่วงที่กำลังเริ่มเข้าสู่ระบบปฏิบัติการ
- The Kernel ส่วนนี้เรียกได้ว่าเปรียบเสมือนคำเรียกของ “Linux” เพราะมันคือระบบส่วนกลางที่ทำหน้าจัดการทรัพยากรต่างๆเช่น CPU, หน่วยความจำและอุปกรณ์ต่อเสริม
- Daemons เป็นส่วนที่ทำงานอยู่เบื้องหลัง (Background Service) เริ่มทำงานตั้งแต่ระหว่างที่ Boot และ เริ่ม Login เข้าสู่ระบบ
- The Shell เป็นคำที่มักจะคุ้นเคยกันสำหรับ Linux เพราะว่า Shell คือการทำงานของคำสั่งที่ทำให้คุณสามารถควบคุมและสั่งการผ่านการพิมพ์ตัวอักษรเข้าไป
- Graphical Server เป็นระบบที่ช่วยเสริมการแสดงผลบนจอ Monitor
- Desktop Environment คือส่วนที่ผู้ใช้งานได้ใช้งานจริง ซึ่งมีให้เลือกได้หลายที่โดยซึ่งก็คือชุดของแอปพลิเคชันต่างๆที่ถูกจำมารวมกัน
- แอปพลิเคชัน เนื่องจาก Desktop Enviroment นั้นไม่ได้จัดแอปพลิเคชันมาครบเหมือน Window หรือ Mac เนื่องจากระบบปฏิบัติการลินุกซ์มีซอฟต์แวร์ที่มีคุณภาพที่ง่ายต้องการค้นหาและติดตั้งระบบปฏิบัติการลินุกซ์ที่ได้รับความนิยมส่วนใหญ่มักจะมีเครื่องมือที่ใช้สำหรับค้นหาและติดตั้งแอปพลิเคชันมาให้ เช่น Ubuntu Linux ก็จะมีศูนย์กลางซอฟต์แวร์คือ apt ที่ใช้ในการดาวน์โหลดและติดตั้งแอปพลิเคชันจากศูนย์กลาง



ภาพที่ 2.1 โครงสร้างของระบบปฏิบัติการลินุกซ์

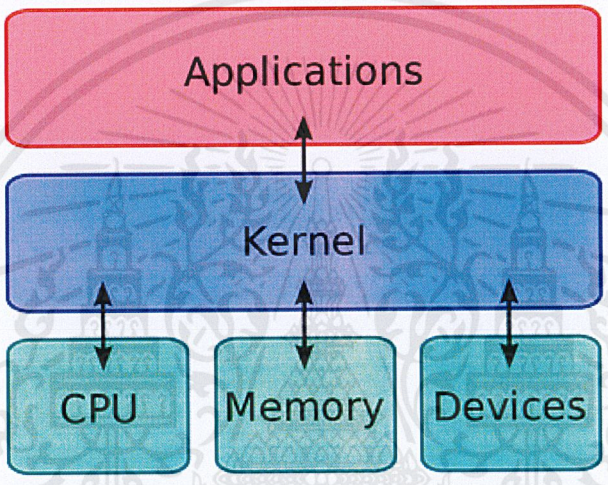
ข้อดีของระบบปฏิบัติการลินุกซ์มีมากมาย ดังนี้

- ลินุกซ์เป็นระบบปฏิบัติการแบบเปิด ที่ผู้ใช้สามารถนำไปใช้ได้อย่างอิสระ และไม่เสียเงิน
- ปราศจากปัญหาเรื่องโปรแกรมเถื่อน เพราะได้รับอนุญาตในการใช้อย่างถูกลิขสิทธิ์
- ไม่ค่อยมีปัญหาเรื่องไวรัส เพราะเป็นระบบปฏิบัติการที่อัพเดทตัวเองอยู่ตลอดเวลา บวกกับมาตรการรักษาความปลอดภัยของลินุกซ์เอง
- เหมาะกับนักพัฒนาด้านโปรแกรมหรือพัฒนาซอฟต์แวร์ที่ชอบคิดค้น สร้างสรรค์ไอเดียใหม่ๆอยู่เสมอ
- มีขั้นตอนการติดตั้งที่ง่าย ไม่ซับซ้อน ใช้เวลาไม่นานและมีความต้องการระบบฮาร์ดแวร์ไม่สูงมาก ที่สำคัญคือมีเสถียรภาพสูงและปลอดภัย
- ผู้ใช้หรือผู้พัฒนาสามารถเปลี่ยนแปลงหน้าตาหรืออินเตอร์เฟซของระบบปฏิบัติการบนหน้าจอ เช่น เปลี่ยนธีมสี เปลี่ยนตำแหน่งเพื่อให้ใช้งานได้อย่างสะดวกตามความต้องการ
- มีเกมแบบถูกลิขสิทธิ์มาให้ได้เลือกเล่นเป็นจำนวนนับร้อยเกม ไม่ว่าจะเป็นเกม 2 มิติ , 3 มิติ ประเภทต่างๆ เช่น เกม Puzzle เป็นต้น
- นำไปติดตั้งใช้งานกับคอมพิวเตอร์เก่าๆ ที่ใช้ระบบปฏิบัติการอื่นไม่ได้แล้ว เหมือนเป็นการนำของเก่ามาใช้ใหม่ เพื่อเป็นการลดขยะอิเล็กทรอนิกส์ และยังได้ใช้งานเครื่องคอมพิวเตอร์เก่าอย่างคุ้มค่าด้วย
- เป็นระบบปฏิบัติการที่สามารถเปิดเครื่องไว้แบบต่อเนื่องได้เป็นเดือนๆ โดยไม่รีสตาร์ทตัวเอง จึงนิยมนำไปใช้เป็นเครื่องเซิร์ฟเวอร์ [3]

2.1.3 Kernel

Kernel คือ โปรแกรมที่เป็นศูนย์กลางในระบบคอมพิวเตอร์ ทำหน้าที่ควบคุมการทำงานตั้งแต่เริ่ม Boot Server รวมถึงการเริ่ม/หยุดโปรแกรม และ อินพุต/เอาต์พุตจากซอฟต์แวร์ทั้งหมด คอยจัดการทรัพยากรต่างๆ หรือ ฮาร์ดแวร์เช่น คีย์บอร์ด, มอนิเตอร์, เครื่องพิมพ์, สปีคเกอร์ โดย Kernel ทำหน้าที่เชื่อมการทำงานระหว่างซอฟต์แวร์และฮาร์ดแวร์เข้าด้วยกันนั่นเอง

ส่วนที่เป็นโค้ดสำคัญของ Kernel จะถูกโหลดเข้าส่วนหน่วยความจำที่จองไว้ให้เฉพาะ เพื่อป้องกันการเขียนทับจากแอปพลิเคชันอื่น การเชื่อมต่อของ Kernel เป็นระดับต่ำสุดที่เกี่ยวข้องกับฮาร์ดแวร์ เมื่อมีการร้องขอไปยัง Kernel เราจะเรียกว่า System Call และส่วนที่ใช้งานจะเรียกว่า Resource



ภาพที่ 2.2 การเชื่อมการทำงานระหว่างซอฟต์แวร์และฮาร์ดแวร์ของ Kernel

หน้าที่หลักๆของ Kernel คือทำงานเป็นสื่อกลางในการเข้าถึงทรัพยากรของระบบ เช่น

1. Central Processing Unit

ทำหน้าที่ควบคุมจัดการโปรแกรมที่กำลังทำงาน โดย Kernel จะรับผิดชอบในการตัดสินใจว่าโปรแกรมแต่ละตัวจะจองหน่วยประมวลผลคอร์ไหนและกี่คอร์ในการทำงาน

2. Random-access Memory

ใช้ในการเก็บข้อมูลของโปรแกรมที่ใช้งาน ซึ่งโดยปกติจะมีโปรแกรมจำนวนมากเข้ามาใช้งานตลอดเวลาตามความต้องการของแต่ละแอปพลิเคชัน ซึ่ง Kernel มีหน้าที่ตัดสินใจว่าหน่วยความจำส่วนไหนที่ Process แต่ละอันสามารถใช้งานได้ และ ควรทำอย่างไรเมื่อหน่วยความจำไม่เพียงพอ

3. Input/Output(I/O) Devices

I/O ของแต่ละอุปกรณ์ เช่น คีย์บอร์ด, เมาส์, ดิสก์, เครื่องพิมพ์ หรือ จอมอนิเตอร์ ทั้งหมดนี้ Kernel จะควบคุมการสื่อสารระหว่างแอปพลิเคชันและฮาร์ดแวร์ให้

Kernel แบ่งออกเป็น 3 ประเภท ดังนี้

1. Monolithic Kernels

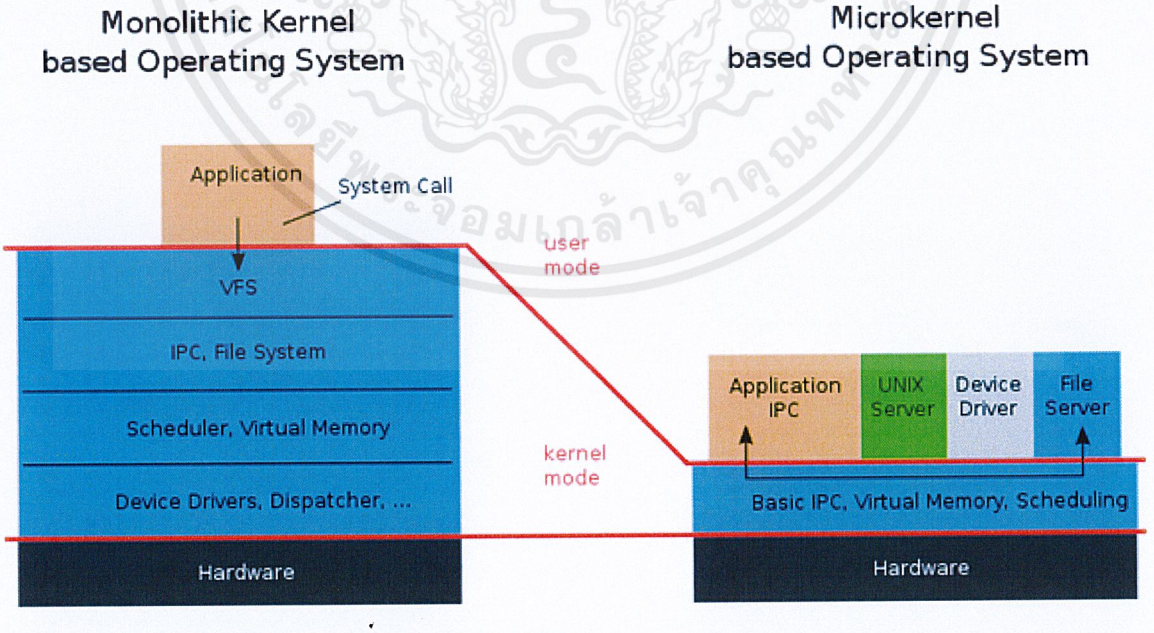
เกิดขึ้นในยุคเริ่มแรกของ Kernel โดยระบบพื้นฐานทั้งหมด เช่น การดำเนินการและการจัดการหน่วยความจำจะถูกรวมอยู่ใน Module เดียวกันภายใน Kernel ซึ่งเป็นผลทำให้ Kernel มีขนาดใหญ่ และ ยากต่อการดูแล ภายหลังจึงได้มีการแยก Module ออกมาและทำการเลือกโหลดใช้งานตามความเหมาะสม เป็นเสมือน Extension ให้ OS เลือกใช้ ทำให้ไม่ต้องทำการปิดและคอมไพล์ใหม่ทั้งหมดเมื่อมีการแก้สิ่งบกพร่อง ซึ่งปัจจุบันระบบปฏิบัติการลินุกซ์ออกแบบตาม Monolithic

2. Microkernels

จากปัญหาในเรื่องขนาดของ Kernel ที่โตขึ้นเรื่อยๆของ Monolithic ทำให้มีการแยกส่วนของระบบพื้นฐานเช่น Driver, Protocol Stack, File System ออกมารันข้างนอก ทำให้ลดขนาดของ Kernel ลง และยังเพิ่มความปลอดภัยและเสถียรภาพให้กับ OS อีกด้วย โดยทั้งหมดจะทำงานในส่วนของ User Space และทำงานบนระบบตามการเรียกใช้ของโปรแกรม โดยระบบปฏิบัติการ QNX ออกแบบตาม Microkernel

3. Hybrid Kernels

ถูกนำมาใช้งานกับ OS ระดับ Commercial มีลักษณะคล้าย Microkernel ยกเว้นแต่ว่ามันได้รวมเอาโค้ดเสริมใน Kernel Space มาเพิ่มความสามารถโดยใช้เป็น Extension ให้กับ Microkernel จึงสรุปได้ว่า Hybrid Kernel เป็น Microkernel ที่มีโค้ดเสริมบางอย่างบน Kernel Space ที่ช่วยทำให้ทำงานได้ไวขึ้น [4]



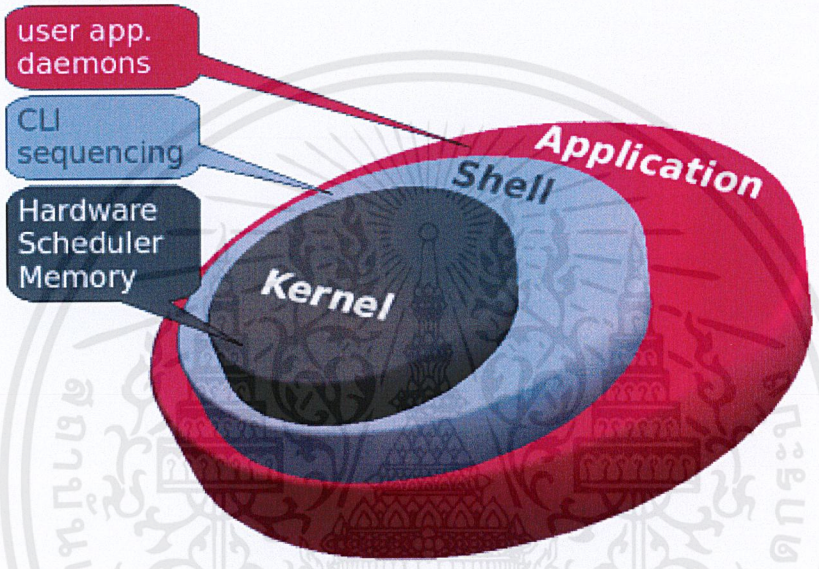
ภาพที่ 2.3 โครงสร้างของ Monolithic Kernels และ Microkernels

2.1.4 Shell

Shell คือตัวแปลงคำสั่งที่ช่วยให้ผู้ใช้งานสามารถเข้าถึงระบบ Operating System มีด้วยกัน 2 แบบคือ

- CLI (Command Line Interface) รับคำสั่งโดยข้อความและแสดงผลในรูปแบบข้อความเช่นกัน
- GUI (Graphical User Interface) รับคำสั่งโดยอาศัยเมาส์และรูปบนจอมอนิเตอร์

ขึ้นอยู่กับระบบคอมพิวเตอร์ที่ใช้งานว่าออกแบบมาสำหรับงานประเภทไหน โดยคำว่า “shell” มีที่มาจากการที่มันทำหน้าที่เป็น Layer ครอบคลุมการทำงานของ Kernel อีกที



ภาพที่ 2.4 ลำดับชั้นการทำงานของ Kernel, Shell และ Application

ส่วนมากแล้ว ตัว Shell ไม่ได้มีการเชื่อมต่อโดยตรงภายใต้ Kernel ถึงแม้ว่าตัว Shell จะสื่อสารกับผู้ใช้งานผ่านทางอุปกรณ์ที่เชื่อมต่อกับคอมพิวเตอร์ แต่ shell ก็ยังคงใช้ Kernel API เหมือนกับแอปพลิเคชันที่ใช้งานกัน โดย Shell จะเข้ามาบริการจัดการระบบโดยอาศัยการแปลงคำสั่งที่ผู้ใช้งานป้อนเข้ามาและนำผลลัพธ์กลับไปแสดง ซึ่งการที่ Shell ทำงานเหมือนแอปพลิเคชัน ทำให้ง่ายต้องการเปลี่ยนแปลงไปใช้แอปพลิเคชันอื่นที่ทำงานเหมือนกัน [5]

2.1.5 Shell Script

Shell Script คือ โปรแกรมหนึ่งบนระบบปฏิบัติการยูนิกซ์ ลินุกซ์ ซึ่ง shell script ทำหน้าที่เป็นส่วนติดต่อผู้ใช้ระหว่างผู้ใช้กับระบบปฏิบัติการยูนิกซ์/ลินุกซ์ ซึ่ง **Shell** ไม่ได้เป็นส่วนหนึ่งของ Kernel แต่ใช้ Kernel ในการประมวลผล ผู้ใช้สามารถสั่งงานระบบปฏิบัติการได้โดยผ่านทาง Secure Shell เท่านั้น โปรแกรม Secure Shell ยังมีคุณสมบัติของ Shell Programming Language ทำให้ผู้ใช้สามารถนำคำสั่งต่างๆ ของ Shell มาเขียนโปรแกรมเก็บเป็นไฟล์ไว้ได้ เรียกว่า “Shell Script”

ประเภทของ Shell ที่นิยมใช้ในปัจจุบัน มีดังนี้

- Bourne Shell (/bin/sh) เป็น Shell ในยุคแรกๆ ที่มีการใช้กันอย่างแพร่หลาย มีการกำหนดโครงสร้างภาษาคล้ายๆกับภาษาอัลกอ สามารถเขียนเป็น Shell Script ได้ และยังเป็นเชลล์มาตราฐานที่มีในระบบปฏิบัติการยูนิกซ์ทุกตัวและยังสามารถย้าย Shell Script ไปยังยูนิกซ์ระบบอื่นได้โดยไม่ต้องแก้ไขสคริปต์ โดย Bourne Shell จะมี Default Prompt เป็นเครื่องหมาย “$”
- C shell (/bin/csh) เป็น Shell ที่พัฒนาขึ้นมาหลังจาก Bourne Shell มีรูปแบบคำสั่งและไวยากรณ์เหมือนกับภาษาซี (C Language) มีฟังก์ชันการทำงานหลากหลาย สะดวก อีกทั้งยังสามารถควบคุมการไหลของข้อมูลได้ดีกว่า Bourne Shell และยังมีความสามารถในการเรียกใช้คำสั่งที่ใช้ไปแล้ว โดยC shell จะมี Default Prompt เป็นเครื่องหมาย “%”
- Korn Shell (/bin/ksh) เป็น Shell ที่พัฒนามาจากต้นแบบของ Bourne Shell และ C Shell สามารถทำงานในฟังก์ชันของ Bourne Shell ได้ทุกอย่าง การเขียน Shell Script ทำได้ง่าย และรัดกุมขึ้น สามารถนำคำสั่งที่ใช้ไปแล้วกลับมา Execute ใหม่ได้ ถือได้ว่า Korn Shell เป็นการรวมเอาข้อดีของ Bourne Shell และ C Shell เข้ามาไว้ด้วยกัน แต่ไม่ได้มีในระบบปฏิบัติการยูนิกซ์ทุกตัว โดย Korn Shell จะมี Default Prompt เป็นเครื่องหมาย “$”
- Bourne again shell หรือ bash (แบช) /bin/bash หรือ /usr/local/bin/bash เป็นการเอา Bourne shell กลับมาพัฒนาใหม่ ทำให้สามารถทำงานแบบ Line Editing ได้ และยังได้เพิ่มประสิทธิภาพในการทำงานอีกหลายอย่าง Bash Shell นี้ไม่ใช่มาตรฐานของ Unix Shell แต่เป็น Default shell ของลินุกซ์ในปัจจุบัน โดย Bash จะมี Default Prompt เป็นเครื่องหมาย “# หรือ $” [6]

ความสามารถของ Shell Script คือ

1. เป็นตัวย่อคำสั่ง

Shell Script ช่วยให้ง่ายต่อการใช้งานระบบมากขึ้น ด้วยการกำหนดค่า Environment รวมถึงคำสั่งต่างๆที่ต้องทำ รวมเข้าไปไว้ใน Script เดียวแต่ยังทำงานเหมือนคำสั่งยูนิกซ์ปกติ

2. รวบรวมชุดคำสั่ง

งานที่ต้องรันคำสั่งซ้ำๆกันหลายครั้ง หรือ Batch Job สามารถรวบรวมเข้าเป็น Shell Script เพื่อให้ระบบค่อยๆทำงานทีละชุดคำสั่งไปเรื่อยๆเอง แทนที่จะต้องมาคอยรันคำสั่งเองที่ละบรรทัด

3. การใช้งานทั่วไป

นอกจาก Batch Job แล้ว เราสามารถนำ Shell Loop มาทำงานให้มีความสะดวกยิ่งขึ้น

หลังจากทราบความสามารถของ Shell Script ไปแล้ว จะเห็นว่าข้อดีหลักๆคือการใช้งานค่อนข้างง่ายเพราะทำงานด้วยคำสั่งแบบเดียวกับ Command Line ที่ใช้งาน รวมถึงนักพัฒนาเองก็ไม่ต้องสับสนในเรื่องการใช้ Syntax และยังง่ายต้องการ Debug แต่ก็มีข้อเสียตามมา เช่น การทำงานที่ผิดพลาดหรือไม่ถูกต้องตามความต้องการจากการพิมพ์ผิด ยกตัวอย่าง เช่นการใส่คำสั่งลบ Directory ใน Path ที่ทำงานอยู่ ลงไปใน Shell Script ด้วย rm -rf * / ซึ่งในความเป็นจริงเราต้องการที่จะใช้ rm -rf */ จะเห็น

ว่าต่างกันเพียงแค่มี Space เพิ่มขึ้นมาแต่ผลลัพธ์กลับกลายเป็นการลบ Directory ทิ้งบน Root ทั้งหมด นำมาซึ่งความเสียหายที่ใหญ่หลวง [7]

2.1.6 JavaScript

JavaScript คือ ภาษาคอมพิวเตอร์สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ตที่กำลังได้รับความนิยมอย่างสูง JavaScript เป็นภาษาสคริปต์เชิงวัตถุที่มีเป้าหมายในการออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ต สำหรับผู้เขียนด้วยภาษา HTML สามารถทำงานข้ามแพลตฟอร์มได้ โดยทำงานร่วมกับ ภาษา HTML และภาษาจาวาได้ทั้งทางฝั่งไคลเอนต์ (Client) และ ทางฝั่งเซิร์ฟเวอร์ (Server)

JavaScript ถูกพัฒนาขึ้นโดย Netscape Communications Corporation โดยใช้ชื่อว่า Live Script ออกมาพร้อมกับ Netscape Navigator2.0 เพื่อใช้สร้างเว็บเพจโดยติดต่อกับเซิร์ฟเวอร์แบบ Live Wire ต่อมา Netscape จึงได้ร่วมมือกับบริษัท Sun Microsystem ปรับปรุงระบบของบราวเซอร์เพื่อให้สามารถติดต่อใช้งานกับภาษาจาวาได้ และได้ปรับปรุง Live Script ใหม่เมื่อ ปี 2538 แล้วตั้งชื่อใหม่ว่า JavaScript สามารถทำให้การสร้างเว็บเพจ มีลูกเล่น ต่าง ๆ มากมาย และยังสามารถโต้ตอบกับผู้ใช้ได้อย่างทันที เช่น การใช้เมาส์คลิก หรือ การกรอกข้อความในฟอร์ม เป็นต้น

เนื่องจาก JavaScript ช่วยให้ผู้พัฒนา สามารถสร้างเว็บเพจได้ตรงกับความต้องการ และมีความน่าสนใจมากขึ้น ประกอบกับเป็นภาษาเปิด ที่ใครก็สามารถนำไปใช้ได้ ดังนั้นจึงได้รับความนิยมเป็นอย่างสูง มีการใช้งานอย่างกว้างขวาง รวมทั้งได้ถูกกำหนดให้เป็นมาตรฐานโดย ECMA การทำงานของ JavaScript จะต้องมีการแปลความคำสั่ง ซึ่งขั้นตอนนี้จะถูกจัดการโดยบราวเซอร์ (เรียกว่าเป็น Client-Side Script) ดังนั้น JavaScript จึงสามารถทำงานได้ เฉพาะบนบราวเซอร์ที่สนับสนุน ซึ่งปัจจุบันบราวเซอร์เกือบทั้งหมดก็สนับสนุน JavaScript แล้ว อย่างไรก็ดี สิ่งที่ต้องระวังคือ JavaScript มีการพัฒนาเป็นเวอร์ชันใหม่ๆออกมาด้วย ดังนั้น ถ้านำโค้ดของเวอร์ชันใหม่ ไปรันบนบราวเซอร์รุ่นเก่าที่ยังไม่สนับสนุนก็อาจจะทำให้เกิดข้อผิดพลาดได้

ความสามารถของ JavaScript มีมากมาย เช่น

1. JavaScript ทำให้สามารถใช้เขียนโปรแกรมแบบง่ายๆได้ โดยไม่ต้องพึ่งภาษาอื่น
2. JavaScript มีคำสั่งที่ตอบสนองกับผู้ใช้งาน เช่นเมื่อผู้ใช้คลิกที่ปุ่ม หรือ Checkbox ก็สามารถสั่งให้เปิดหน้าใหม่ได้ ทำให้เว็บไซต์ของเรามีปฏิสัมพันธ์กับผู้ใช้งานมากขึ้น นี่คือข้อดีของ JavaScript เลยก็ว่าได้ที่ทำให้เว็บไซต์ดังๆทั้งหลายเช่น Google Map ต่างหันมาใช้
3. JavaScript สามารถเขียนหรือเปลี่ยนแปลง HTML Element ได้ นั่นคือสามารถเปลี่ยนแปลงรูปแบบการแสดงผลของเว็บไซต์ได้ หรือหน้าแสดงเนื้อหาสามารถซ่อนหรือแสดงเนื้อหาได้แบบง่ายๆนั่นเอง
4. JavaScript สามารถใช้ตรวจสอบข้อมูลได้ สังเกตว่าเมื่อเรากรอกข้อมูลบางเว็บไซต์ เช่น Email เมื่อเรากรอกข้อมูลผิดจะมีหน้าต่างฟ้องขึ้นมาว่าเรากรอกผิด หรือลืมกรอกอะไรบางอย่าง เป็นต้น

5. JavaScript สามารถใช้ในการตรวจสอบผู้ใช้ได้เช่น ตรวจสอบว่าผู้ใช้ ใช้ Web Browser อะไร
6. JavaScript สร้าง Cookies (เก็บข้อมูลของผู้ใช้ในคอมพิวเตอร์ของผู้ใช้เอง) ได้

ข้อดีและข้อเสียของ JavaScript คือ การทำงานของ JavaScript เกิดขึ้นบนบราวเซอร์ ดังนั้นไม่ว่าจะใช้เซิร์ฟเวอร์อะไร หรือที่ไหน ก็ยังคงสามารถใช้ JavaScript ในเว็บเพจได้ ต่างกับภาษาสคริปต์อื่น เช่น Perl, PHP หรือ ASP ซึ่งต้องแปลความและทำงานที่ตัวเครื่อง ดังนั้นจึงต้องใช้บนเซิร์ฟเวอร์ที่สนับสนุนภาษาเหล่านี้เท่านั้น อย่างไรก็ดี จากลักษณะดังกล่าวก็ทำให้ JavaScript มีข้อจำกัดคือไม่สามารถรับและส่งข้อมูลต่างๆกับเซิร์ฟเวอร์โดยตรง เช่น การอ่านไฟล์จากเซิร์ฟเวอร์ เพื่อนำมาแสดงบนเว็บเพจ หรือรับข้อมูลจากผู้ชม เพื่อนำไปเก็บบนเซิร์ฟเวอร์ เป็นต้น ดังนั้นงานลักษณะนี้ จึงยังคงต้องอาศัยภาษา Server-Side Script อยู่ [8]

2.1.7 NodeJS

NodeJS เป็นเทคโนโลยีฝั่ง Server Side ที่ถูกพัฒนาด้วยภาษา JavaScript เดิมทีภาษา JavaScript ทำงานฝั่ง Client เป็นหลัก แต่จริงๆแล้ว NodeJS เป็น Client หรือ Server ก็ขึ้นอยู่กับจุดประสงค์ของแอปพลิเคชันนั้น แต่จุดตั้งต้นเริ่มมาจาก Server Side เป็นหลัก ผู้สร้าง คือ Ryan Dahl ซึ่ง NodeJS คือ JavaScript ที่มีการ Complied เป็น Byte Code ด้วย V8 Engine ของ Google และ Debug ได้ ต่างจาก JavaScript ในยุคแรกๆ ทำให้แก้ปัญหาได้ง่ายขึ้น ทำงานได้บนทุกระบบปฏิบัติการยอดนิยม และมาพร้อมกับเทคโนโลยีที่เรียกว่า Non - Blocking I/O [9]

การทำงานของ Node เรียกว่า การขับเคลื่อนด้วย Event ต่างๆที่เกิดขึ้น ทำให้เรากระโดดจาก Event หนึ่งที่เสร็จแล้วไปอีก Event หนึ่งได้ด้วยการสั่งงานมันต่อเนื่องกันไปเรื่อยๆ หรือว่าการสั่งให้หลายๆ Event เริ่มทำงานในเวลาใกล้ๆเคียงกันเลยก็ได้เช่นกัน ประโยชน์อีกอย่างที่ได้จาก Event Driven ก็คือ การสั่งให้มันรอรับ Event นั้นไปตลอดการณ์ โดยไม่เปลืองทรัพยากร เช่น การเชื่อมต่อไปยัง Streaming Channel สักที่หนึ่ง อาจจะเป็น Text หรือข้อมูลบางอย่างเช่นปริมาณน้ำฝนทิ้งเอาไว้ หากต้นทางของ Streaming ยังไม่มีข้อมูลส่งมา มันก็จะไม่เกิด Event ใดๆ node.js ก็จะอยู่นิ่งๆ แต่หากต้นทาง Streaming มาแล้ว node.js ก็จะทำงานเพื่อตอบสนองต่อ Event ที่เกิดขึ้นนั้นทันที

ด้วยประโยชน์ของ Event Driven นี้ทำให้เราเอามาต่อยอดได้อีกหลายอย่างเช่น การ Subscribe Pubsub (เช่น Pubsub ใน Redis) หากเกิด Event Publish เมื่อไร node.js ที่ Subscribe รอเอาไว้อยู่นานแล้ว ก็จะถูกกระตุ้นแล้วทำงานตามที่เราเขียนเอาไว้โดยทันที ซึ่งแบบนี้ จะทำให้ลดการสูญเสีย Header ในการเริ่มต้นประมวลตั้งแต่จุดแรกไปได้มาก รวมทั้ง ไม่ต้องไปหน่วงเวลาเพื่อคอยเช็ค Event เหมือนเวลาเราเขียน ajax เลยแม้แต่น้อย

ในปัจจุบัน NodeJS ถูกนำมาทำเป็น Web Server, Mobile Hybrid, IOT, Webkit, TVOS, OS และอื่นๆอีกมาก เรียกได้ว่าเข้าถึงได้หลากหลายเทคโนโลยี ซึ่งเหตุผลที่ NodeJS ได้รับความนิยมหลักๆคือ เขียนโค้ดเข้าใจได้ง่าย มี Library ฟรีมากมาย ใช้ทรัพยากรน้อย และเรียนรู้ได้เร็วสำหรับโปรแกรมเมอร์ทุกๆระดับ [10]

บทที่ 3
วิธีดำเนินการวิจัย

รายงานสหกิจฉบับนี้เป็นการพัฒนาโปรแกรมสำหรับการประมวลผลข้อมูล ซึ่งในบทนี้จะกล่าวถึงขั้นตอนการดำเนินงาน รวมถึงการวิเคราะห์และออกแบบโครงสร้างและการทำงานของระบบ โดยมีรายละเอียด ดังนี้

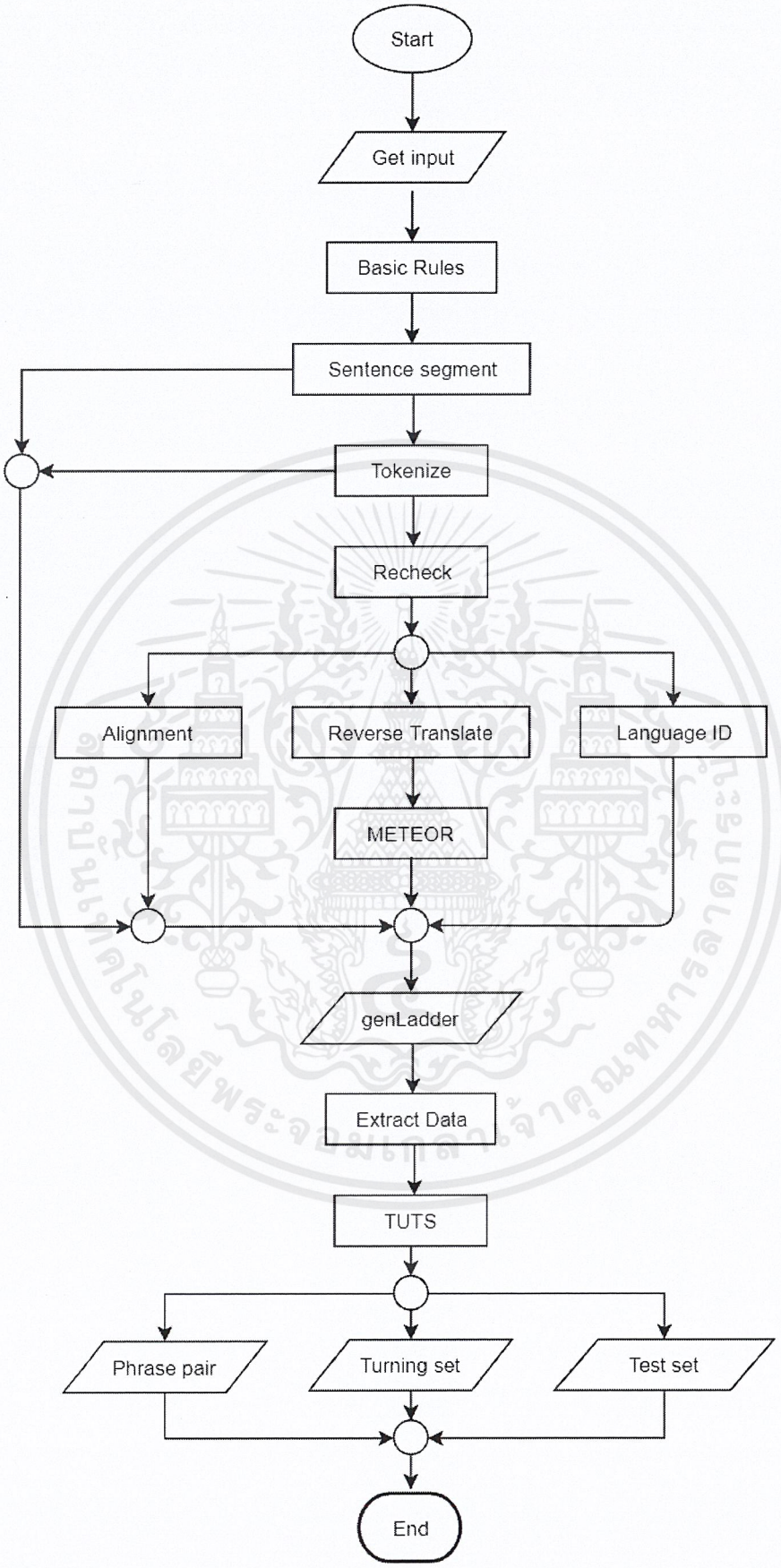
3.1 ขั้นตอนการดำเนินงาน

1. ศึกษาความต้องการและขอบเขตของโครงงาน
2. วิเคราะห์ความต้องการและวางแผนตารางการพัฒนาโครงงานให้สอดคล้องกับงาน เพื่อให้โครงงานสำเร็จภายในระยะเวลาที่กำหนด
3. เลือกเทคโนโลยีสำหรับพัฒนาโครงงานให้เหมาะสมกับงาน
4. ศึกษาเทคโนโลยีที่เลือกใช้สำหรับการพัฒนาโครงงาน
5. ศึกษาและทำความเข้าใจเกี่ยวกับกระบวนการทำงานของเฟรมเวิร์คที่บริษัทใช้ในการประมวลผลข้อมูล และหน้าที่การทำงานของทีม Data Production
6. วิเคราะห์กระบวนการทำงานของเฟรมเวิร์คเพื่อหาจุดบกพร่องหรือจุดที่สามารถปรับปรุงเพื่อให้มีการทำงานที่ดีขึ้นหรือช่วยลดระยะเวลาการทำงานได้
7. พัฒนาโปรแกรมที่สามารถช่วยลดข้อผิดพลาดของกระบวนการทำงานของเฟรมเวิร์คหรือช่วยลดระยะเวลาในการทำงานของทีมประมวลผลข้อมูลได้
8. ทดสอบและปรับปรุงการทำงานของโปรแกรมให้มีขั้นตอนการทำงานที่ถูกต้อง เหมาะสมและมีประสิทธิภาพมากขึ้น
9. สรุปผลและจัดทำเอกสารอธิบายการทำงานและโครงสร้างของโครงงาน

3.2 ภาพรวมการทำงานของเฟรมเวิร์ค (Drop Folder) และโปรแกรมที่พัฒนา

3.2.1 ภาพรวมการทำงานของเฟรมเวิร์ค (Drop Folder)

บริษัทมีการใช้เฟรมเวิร์คที่พัฒนาโดยนักพัฒนาของบริษัทเอง เนื่องจากเฟรมเวิร์คที่มีให้ใช้ทั่วไปยังไม่ตรงกับความต้องการของบริษัทมากพอ นอกจากนี้การพัฒนาเฟรมเวิร์คขึ้นมาใช้เองยังทำให้บริษัทสามารถปรับแต่งหรือแก้ไขเฟรมเวิร์คของตนเองได้ง่าย และบริษัทก็ยังขายผลิตภัณฑ์เฟรมเวิร์คนี้อีกด้วย แต่จะขายให้กับลูกค้าบางรายเท่านั้น ซึ่งเฟรมเวิร์คนี้มีชื่อเรียกกันว่า Drop Folder โดยมีการทำงานตามโครงสร้าง ดังนี้



ภาพที่ 3.1 ภาพรวมการทำงานของเฟรมเวิร์ค Drop Folder

จากรูปที่ 3.1 ไฟล์อินพุตที่รับมาจะเป็นไฟล์ .tab ซึ่งหมายถึงไฟล์ที่มีการจับคู่ภาษากันมาแล้ว โดยคั่นระหว่างคู่ภาษาด้วย tab ซึ่งคู่ภาษาในไฟล์อินพุตไม่จำเป็นต้องเป็นคู่ที่แปลภาษาอย่างถูกต้อง โดยไฟล์อินพุตสามารถนำมาได้จาก 4 แหล่งข้อมูล คือ การดึงข้อมูลจากเว็บไซต์, การนำข้อมูลมาจากคลังข้อมูลของบริษัท หรือที่เรียกว่า Industry Data, การนำข้อมูลมาจากสมาคม หรือที่เรียกว่า Opus Data และการนำข้อมูลมาจากลูกค้า ซึ่งไฟล์อินพุตที่จะถูกส่งเข้ากระบวนการ Basic Rules ไม่ว่าจะได้มาจากวิธีใด จะต้องนำมารวมกับข้อมูลจากคลังข้อมูลของบริษัทเสมอเพื่อให้ได้ผลลัพธ์ที่ถูกต้องมากยิ่งขึ้น โดยเฟรมเวิร์ค Drop Folder มีการทำงานตามลำดับขั้น ดังนี้

1. Basic Rules

   ขั้นตอนนี้เป็นการทำความสะอาดข้อมูลเบื้องต้นของไฟล์ที่ได้รับมา ยกตัวอย่างเช่น การลบ tag ของไฟล์ html, การตรวจสอบรูปแบบของไฟล์, การแปลงภาษาจีนเป็นแบบ Simplified (ตัวอักษรจีนที่ถูกย่อให้เส้นขีดน้อยลงและง่ายต่อการจำ) หากไฟล์ที่ได้รับมาเป็นภาษาจีนแบบ Traditional (ตัวอักษรจีนแบบดั้งเดิม), การทำ Detokenize และการแบ่งไฟล์อินพุตที่มีขนาดใหญ่ให้เป็นไฟล์ย่อย ไฟล์ละ 10,000 บรรทัด เป็นต้น

1. Sentence Segment

   ขั้นตอนนี้เป็นการตัดข้อความยาวๆให้เป็นประโยค เพื่อให้ง่ายต่อการตัดคำ โดยแต่ละภาษาก็จะมีอัลกอริทึมในการตัดข้อความที่แตกต่างกันออกไป

2. Tokenize

   ขั้นตอนนี้เป็นการนำข้อความยาวๆมาแตกเป็นคำ หรือที่เรียกว่า การตัดคำ โดยแต่ละภาษาก็จะมีอัลกอริทึมในการตัดคำที่แตกต่างกันออกไป

3. Recheck

   ขั้นตอนนี้เป็นการตรวจสอบไฟล์หลังจากผ่านขั้นตอน Tokenize เช่น ตัวอักษรในไฟล์มีการเข้ารหัสเป็นแบบ UTF-8 (การบันทึกอักษรอังกฤษเป็นแบบ 8 บิต และบันทึกอักษรไทยแบบ 24 บิต) หรือไม่ หากไม่ใช่ ต้องทำการแปลงเป็นแบบ UTF-8 ก่อน ซึ่งเหตุผลที่ต้องมีการตรวจสอบอีกครั้งหลังจากการ Tokenize เนื่องจากเคยมีเหตุการณ์ที่ไฟล์อินพุตก่อนการ Tokenize มีการเข้ารหัสแบบ UTF-8 แต่เมื่อผ่านขั้นตอน Tokenize กลับกลายเป็นการเข้ารหัสแบบอื่น ซึ่งจากข้อสันนิษฐานคิดว่าน่าจะเกิดจากข้อผิดพลาดของตัว Tools ที่ทางบริษัทใช้ในการทำ Tokenize ทำให้ขั้นตอนต่อไปก็ผิดพลาดไปด้วย ทำให้ต้องมีการตรวจสอบอีกครั้งก่อนผ่านไปสู่ขั้นตอนต่อไป เป็นต้น

4. Alignment

   ขั้นตอนนี้เป็นการตรวจสอบความหมายของคู่ภาษา ว่ามีความหมายเหมือนกันคิดเป็นคะแนนเท่าไรจาก 0 ถึง 1 และคิดเป็นกี่เปอร์เซ็นต์ เช่น ประโยค “I want to go home” และประโยค “ฉันอยากกลับบ้าน” มีความหมายเหมือนกันคิดเป็น 100 เปอร์เซ็นต์ เป็นต้น

5. Language identification

ขั้นตอนนี้เป็นการระบุภาษาของข้อความ โดยจะทำการตรวจสอบว่าข้อความนั้นประกอบด้วยภาษาใดบ้างและภาษานั้นปรากฏอยู่กี่เปอร์เซ็นต์ของข้อความ เช่น ประโยค “ฉัน want กลับบ้าน” ประกอบด้วยภาษาไทยและภาษาอังกฤษ โดยมีภาษาไทยอยู่ที่ 90 เปอร์เซ็นต์ และภาษาอังกฤษ 10 เปอร์เซ็นต์ เป็นต้น

6. Reverse translate และ METEOR

ขั้นตอนนี้เป็นการนำภาษาฝั่งปลายทางมาแปลให้ได้ภาษาฝั่งต้นทางอีกครั้ง และนำภาษาฝั่งต้นทางเดิมและภาษาฝั่งต้นทางที่แปลใหม่มาเปรียบเทียบกันว่ามีความหมายเหมือนกันคิดเป็นคะแนนเท่าไรจาก 0 ถึง 1 และคิดเป็นกี่เปอร์เซ็นต์ เช่น ประโยค “I want to go home” และประโยค “ฉันอยากกลับบ้าน” เมื่อนำประโยค “ฉันอยากกลับบ้าน” มาแปลภาษาอีกครั้งได้ผลลัพธ์ว่า “I live at home” จากนั้นนำประโยค “I want to go home” และ “I live at home” มาเปรียบเทียบกัน มีความหมายเหมือนกันคิดเป็น 20 เปอร์เซ็นต์ เป็นต้น ซึ่งการทำงานของขั้นตอนนี้ใช้เทคนิคของ METEOR

7. Generate Ladder

ขั้นตอนนี้เป็นการเก็บรวบรวมผลลัพธ์ที่ได้จากขั้นตอนการทำ Sentence Segment, Tokenize, คะแนนจากการทำ Alignment, คะแนนจากการทำ Language Identification และคะแนนจากการทำ Reverse Translate มาเก็บไว้ด้วยกันเป็นไฟล์ประเภท XML ซึ่งสามารถนำมาเป็นไฟล์อินพุตในครั้งต่อไปได้

8. Extract Data

ขั้นตอนนี้เป็นกระบวนการสกัดข้อมูลที่อยู่ในระดับความพึงพอใจที่รับได้ออกมา โดยทำการสกัดข้อมูลด้วยเงื่อนไขต่างๆมากมาย เช่น เลือกเฉพาะคู่ภาษาที่มีคะแนนจากการทำ Alignment มากกว่าหรือเท่ากับ 80 เปอร์เซ็นต์, เลือกเฉพาะคู่ภาษาที่มีคะแนนจากการทำ Reverse Translate มากกว่าหรือเท่ากับ 80 เปอร์เซ็นต์, เลือกเฉพาะคู่ภาษาที่มีคะแนนจากการทำ Language Identification มากกว่าหรือเท่ากับ 80 เปอร์เซ็นต์ และเลือกเฉพาะคู่ภาษาที่มีจำนวนคำของทั้งสองฝั่งภาษาต่างกันไม่เกิน 4 คำ เป็นต้น ซึ่งเงื่อนไขเหล่านี้จะแตกต่างกันไปตามแต่ละภาษา

9. TUTS

ขั้นตอนนี้เป็นการสุ่มคู่ภาษาจากผลลัพธ์ที่ได้จากการสกัดข้อมูลออกมา โดยนำออกมาจำนวน 2,000 คู่ภาษาสำหรับ Turning Set (TU) และ 1,000 คู่ภาษาสำหรับ Test Set (TS) ซึ่งความหมายของ Turning Set และ Test Set จะอธิบายในลำดับถัดไป

10. Phrase Pair (PP)

เป็นคู่วลี หรือคู่ภาษาที่ได้มาจากกระบวนการสกัดข้อมูล โดยเอาคู่ภาษาที่เป็น Turning Set และ Test Set แยกออกจาก Phase Pair หรือคู่วลี เพื่อนำไปใช้ในการส่งเทรนหรือสอนต่อไป

11. Turning Set (TU)

เป็นคู่วลีที่แยกออกมาเพื่อเตรียมไว้ใช้ในการปรับความถูกต้องของการแปลภาษาหากคะแนนจากการประเมินผลหลังผ่านการส่งเทรนหรือสอนแล้วไม่อยู่ในระดับที่น่าพึงพอใจ

12. Test Set (TS)

เป็นคู่วลีที่แยกออกมาเพื่อเตรียมไว้ใช้ในการตรวจสอบความถูกต้องของการแปลภาษาของโมเดลที่ผ่านการส่งเทรนหรือสอนมาแล้ว

3.2.2 ภาพรวมการทำงานของโปรแกรมที่พัฒนา

โปรแกรมนับคำ ประโยค บรรทัด ย่อหน้าและรูปภาพในไฟล์ พัฒนาโดยภาษาจาวา ช่วยให้เราสามารถนับสิ่งต่างๆที่ต้องการในไฟล์ได้สะดวกมากยิ่งขึ้น

โปรแกรมตรวจสอบจำนวนคู่ภาษา พัฒนาโดยภาษาจาวา ช่วยในการตรวจสอบจำนวนคู่ภาษาและสามารถแก้ไขได้ก่อนหากมีข้อความใดในภาษาฝั่งใดฝั่งหนึ่งไม่มีคู่

โปรแกรมตรวจสอบอายุของไฟล์และวันที่มีการปรับปรุงไฟล์ล่าสุด พัฒนาโดย Shell Script ช่วยให้เราตรวจสอบไฟล์ได้ง่ายขึ้น

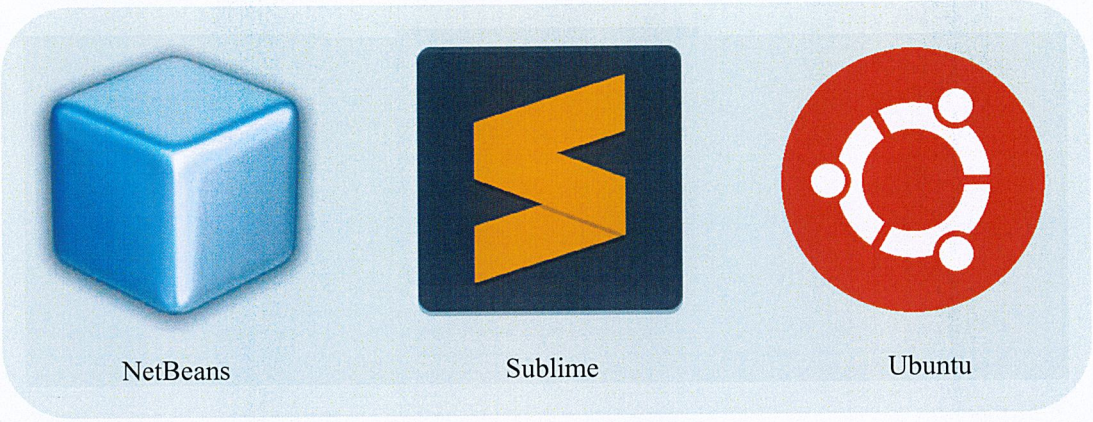
โปรแกรมตรวจสอบการทำงานของเครื่อง Server ที่ใช้ในการทำงานในระบบ พัฒนาโดย Shell Script , JacaScript และ Node JS ช่วยทำการตรวจสอบการทำงานของเครื่องเซิร์ฟเวอร์ในการทำ Sentence Segment และ Tokenize และแจ้งเตือนหากมีข้อผิดพลาดเกิดขึ้น

โปรแกรมทำความสะอาดข้อมูลก่อนนำไปประมวลผล พัฒนาโดย Shell Script ช่วยให้เราสามารถทำความสะอาดข้อมูลได้ง่ายและสะดวกมากยิ่งขึ้น

โปรแกรมตรวจสอบและหาชื่อไฟล์ที่ยังไม่ถูกประมวลผล ช่วยให้เราตรวจสอบไฟล์ที่ยังไม่ถูกประมวลผลจากข้อผิดพลาดบางอย่างได้สะดวกมากขึ้น เนื่องจากการทำงานเช่นนี้ มักจะมีไฟล์จำนวนมาก ซึ่งการตรวจสอบด้วยมนุษย์เองถึงแม้จะทำได้แต่อาจเกิดข้อผิดพลาดได้ง่าย

โปรแกรมแยกข้อความในไฟล์โดยภาษา พัฒนาโดย Shell Script ซึ่งมีประโยชน์อย่างมากเมื่อบริษัทได้ไฟล์มาจากลูกค้าที่ไม่ได้แยกภาษามาให้ เนื่องจากก่อนการประมวลผลใดๆเราต้องทำการแยกฝั่งภาษาก่อน ซึ่งลูกค้าแต่ละรายก็จะมีไฟล์ต้นฉบับมาให้มากมาย การแยกไฟล์เองโดยมนุษย์สามารถทำได้แต่ต้องใช้เวลานาน โปรแกรมนี้จึงช่วยลดระยะเวลาในการประมวลผลได้เป็นอย่างมาก

โปรแกรมตรวจสอบและลบข้อความภาษาจีนแบบดั้งเดิม (Traditional) ออกจากไฟล์ พัฒนาโดย Shell Script ช่วยให้เราแยกภาษาจีนแบบดั้งเดิมออกจากไฟล์ได้สะดวกมากยิ่งขึ้น



ภาพที่ 3.2 เครื่องมือที่ใช้สำหรับพัฒนาโปรแกรม

ภาพที่ 3.3 ภาษาที่ใช้สำหรับพัฒนาโปรแกรม

3.3 การออกแบบโครงสร้างของโปรแกรม

3.3.1 การนับคำ ประโยค บรรทัด ย่อหน้าและรูปภาพในไฟล์

โปรแกรมนับคำ ประโยค บรรทัด ย่อหน้าและรูปภาพในไฟล์ ถูกพัฒนาโดยภาษาจาวา เนื่องจากภาษาจาวามี Library สำหรับอ่านไฟล์ได้หลากหลายรูปแบบ ซึ่งครอบคลุมทั้ง 5 ประเภทของไฟล์ที่ต้องการ คือ Document (.doc และ .docx), Power point (.ppt และ .pptx), Excel (.xls และ .xlsx), Text และ Portable Document Format (PDF) ซึ่งสามารถเรียกใช้ได้ง่าย นอกจากนี้โปรแกรมนี้ยังจะต้องนำไปเป็นส่วนหนึ่งของ LSscript ซึ่งเป็นอีกหนึ่งผลิตภัณฑ์ที่บริษัทขายให้กับลูกค้าอีกด้วย

3.3.2 การตรวจสอบจำนวนคู่ภาษา

โปรแกรมตรวจสอบจำนวนคู่ภาษา ต้องการพัฒนาขึ้นเพื่อใช้ในการตรวจสอบไฟล์อินพุตของลูกค้า กรณีที่บริษัทได้รับไฟล์จากลูกค้าในรูปแบบ XML เพื่อเป็นการตรวจสอบไฟล์เบื้องต้นก่อนนำมาประมวลผล โดยโปรแกรมตรวจสอบจำนวนคู่ภาษาจะได้รับไฟล์อินพุตเป็นไฟล์ประเภท Extensible

Markup Language (XML) ซึ่งภายใน Tag tuv จะมีการแสดงภาษาหรือ Lang ซึ่งระบุว่าข้อความถัดจากนั้นเป็นภาษาใด เพื่อให้บริษัทสกัดข้อมูลของคู่ภาษาออกมาประมวลผลต่อไป

จากเหตุผลข้างต้นจึงทำการออกแบบให้พัฒนาโปรแกรมด้วยภาษาจาวา เนื่องจากภาษาจาวามี Library ที่อ่านไฟล์ประเภท XML และสกัดข้อมูลได้ง่าย และเนื่องจากบริษัทเน้นการทำงานด้วยภาษาจาวาเป็นหลักอีกด้วย

3.3.3 การตรวจสอบอายุของไฟล์และวันที่มีการปรับปรุงไฟล์ล่าสุด

เนื่องจากบริษัทมีการทำงานกับข้อมูลจำนวนมาก ทำให้มีไฟล์มากมายซึ่งแต่ละไฟล์ก็มีประเภทที่ต่างกัน เมื่อมีไฟล์จำนวนมาก บางครั้งอาจมีการแก้ไขหรือปรับปรุงไฟล์เป็นหลายเวอร์ชัน จึงอาจเกิดความสับสนได้ง่ายและด้วยอีกหลายๆเหตุผล แต่เมื่อมีไฟล์และไดเรกทอรีมากมาย การจะเข้าไปตรวจสอบไฟล์แต่ละไฟล์นั้นค่อนข้างยุ่งยาก จึงจำเป็นต้องมีโปรแกรมที่ช่วยในการหาและคำนวณอายุของไฟล์ขึ้น โปรแกรมนี้ถูกออกแบบให้พัฒนาโดย Shell Script เนื่องจากบริษัททำการประมวลผลบน Linux ซึ่ง Shell Script มีความสามารถในการจัดการไฟล์ได้ดีและเป็นภาษาที่พัฒนาง่าย

```
thapanee@DESKTOP-TR7UVSC:/mnt/d/work2.2561/project_coop/script/check_age_and_lastmodified$ cat file.config
path1=/home|pdf
path2=/home|txt
```

ภาพที่ 3.4 ตัวอย่าง Config File เพื่อบอกว่าต้องการหาไฟล์จากไดเรกทอรีใดและเป็นไฟล์ประเภทใด

3.3.4 การตรวจสอบการทำงานของเครื่อง Server ที่ใช้ในการทำงานในระบบ

โปรแกรมตรวจสอบการทำงานของเครื่อง Server ที่ใช้ในการทำงานในระบบ ถูกพัฒนาขึ้นเนื่องจากใน Drop Folder มีขั้นตอนของการทำ Sentence Segment และ Tokenize ซึ่งต้องทำบนเครื่องเซิร์ฟเวอร์จำนวนมาก ซึ่งเมื่อมีไฟล์เข้าไปสู่ขั้นตอน Sentence segment หรือ Tokenize ในขณะที่ตัวโปรแกรมไม่ทำงาน หรือมีการทำงานของโปรแกรมแต่เซอร์วิสที่โปรแกรมเรียกใช้ไม่ทำงาน ทำให้ไฟล์นั้นๆเสียหาย และเกิดการสูญเสียข้อมูล แต่เราจะไม่สามารถทราบว่าเมื่อไรที่จะเกิดเหตุการณ์เหล่านี้หากเราไม่ได้เข้าสู่ระบบเครื่องเซิร์ฟเวอร์นั้น

จากเหตุผลข้างต้นจึงทำการออกแบบให้มีโปรแกรมที่คอยตรวจสอบการทำงานของโปรแกรมและเซอร์วิสที่โปรแกรมเรียกใช้ โดยทำการตรวจสอบทุก 30 นาที หากโปรแกรมหรือเซอร์วิสที่เครื่องเซิร์ฟเวอร์ใดไม่ทำงาน จะทำการส่งอีเมลแจ้งเตือนเพื่อให้ทราบถึงปัญหาและแก้ไขอย่างรวดเร็ว โดยโปรแกรมนี้ถูกออกแบบให้พัฒนาด้วย Shell Script เนื่องจากเป็นภาษาที่พัฒนาง่ายและสามารถเข้าสู่ระบบเครื่องเซิร์ฟเวอร์อื่นได้ง่าย นอกจากนี้ในส่วนของการตรวจสอบ Service ที่ตัว Tool เรียกใช้ในการทำ Sentence Segment หรือ Tokenize จะใช้ NodeJS ในการพัฒนา

```
# ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
#       Tokenize
# ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
frontend front_Tokenize-7020
     bind :7020
     monitor-uri        /haproxy
     default_backend    backend_Tokenize

     log                 global
     option              httplog
     option              dontlognull
backend backend_Tokenize
     mode http
     balance roundrobin
     #BKK
     #server  172.17.101.96 172.17.101.96:7020 weight 1 check
     server  172.17.101.228 172.17.101.228:7020 weight 1 check
     #server  172.17.101.236 172.17.101.236:7020 weight 1 check
     server  172.17.101.238 172.17.101.238:7020 weight 1 check
     server  172.17.101.239 172.17.101.239:7020 weight 1 check
     server  172.17.101.230 172.17.101.230:7020 weight 1 check
     #server  172.17.101.232 172.17.101.232:7020 weight 1 check
     server  172.17.101.234 172.17.101.234:7020 weight 1 check
     server  172.17.101.235 172.17.101.235:7020 weight 1 check
     server  172.17.110.134 172.17.110.134:7020 weight 1 check
     server  172.17.110.140 172.17.110.140:7020 weight 1 check
     server  172.17.110.141 172.17.110.141:7020 weight 1 check
     #server  172.17.110.144 172.17.110.144:7020 weight 1 check


# ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
#       SentenceSegment
# ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
frontend front_SentenceSegment-7050
     bind :7050
     monitor-uri        /haproxy
     default_backend    backend_SentenceSegment

     log                 global
     option              httplog
     option              dontlognull
backend backend_SentenceSegment
     mode http
     balance roundrobin
     #BKK
     server  172.17.101.199 172.17.101.199:7050 weight 1 check
     server  172.17.101.228 172.17.101.228:7050 weight 1 check
     server  172.17.101.239 172.17.101.239:7050 weight 1 check
     #server  172.17.101.232 172.17.101.232:7050 weight 1 check
     server  172.17.101.235 172.17.101.235:7050 weight 1 check
     server  172.17.110.139 172.17.110.139:7050 weight 1 check
     server  172.17.110.141 172.17.110.141:7050 weight 1 check
     #server  172.17.110.144 172.17.110.144:7050 weight 1 check
```

ภาพที่ 3.5 ตัวอย่าง config File ที่แสดงหมายเลขไอพีของเครื่องเซิร์ฟเวอร์ที่มีการทำงานอยู่

3.3.5 การทำความสะอาดข้อมูลก่อนนำไปประมวลผล

โปรแกรมทำความสะอาดข้อมูลก่อนนำไปประมวลผล ต้องการพัฒนาขึ้นเนื่องจากบริษัทต้องการเพิ่ม Dictionary เพื่อให้การแปลภาษามีความถูกต้องมากยิ่งขึ้น ซึ่งการเตรียมข้อมูลบางครั้งมีคำที่ไม่มีความหมาย หรืออักขระพิเศษติดมาจากการดึงข้อมูลจากเว็บไซต์ จึงจำเป็นต้องกำจัดคำเหล่านี้ออกไป ซึ่งรูปแบบที่ต้องการกำจัดก็แตกต่างกันไปในแต่ละภาษาและมีรูปแบบเพิ่มขึ้นตลอดเวลา

จากเหตุผลข้างต้นจึงทำการออกแบบโปรแกรมที่ช่วยให้การกำจัดรูปแบบที่ไม่ต้องการได้ง่ายขึ้น เพียงใส่รูปแบบใน Config File และให้ตัวโปรแกรมตรวจสอบภาษาและกำจัดเพียงรูปแบบที่ตรงกับภาษาที่ต้องการเท่านั้น ทำให้ง่ายต่อการทำความสะอาดข้อมูลมากยิ่งขึ้น โดยโปรแกรมนี้ถูกออกแบบให้พัฒนาด้วย Shell Script เนื่องจากเป็นภาษาที่พัฒนาง่าย

```
thapanee@DESKTOP-TR7DVSC:/mnt/d/work2.2561/project_coop/script/cleanDict/cleanDict$ cat pattern_for_clean.cfg
\\(.*?\\)
\\{.*?\\}
\\[.*?\\]
\\<.*?\\>
\\|
\\®
\\™
EN [\\/]{0,1}sth.
EN sb.\'s
EN sb.
BG нкг.
BG нкм.
BG нщ.
CS č-[o|u]
CS k-[u]{0,1}[c]{0,1}o
CS čem
CS čím
CS kom
CS kým
DA [\\/]{0,1}ng[t|n].
DE [\\/]{0,1}etw.
DE jd[s|m|n]{0,1}.
DE Gen.
DE Dat.
DE Akk.
EO [\\/]{0,1}i[u|o][n]{0,1}
ES [\\/]{0,1}algo
ES [a|u]\/c
ES c\/u
ES alg[o|.|n]
```

ภาพที่ 3.6 ตัวอย่าง Config File เพื่อบอกว่าต้องการลบรูปแบบใดออกจากภาษาใด

จากรูปที่ 3.6 จะเห็นว่าคอลัมน์แรกแสดงภาษาและคอลัมน์ที่สองแสดงรูปแบบที่ต้องการลบออกจากภาษานั้นๆ มีเพียง 9 บรรทัดแรกของ Config File เท่านั้นที่ไม่แสดงภาษา หมายความว่าต้องการลบรูปแบบเหล่านั้นออกจากทุกภาษานั่นเอง

3.3.6 การตรวจสอบและหาชื่อไฟล์ที่ยังไม่ถูกประมวลผล

โปรแกรมตรวจสอบและหาชื่อไฟล์ที่ยังไม่ถูกประมวลผล ถูกพัฒนาขึ้นเนื่องจากการทำงานมีขั้นตอนจำนวนมาก ซึ่งการทำแต่ละขั้นตอนเราควรมั่นใจได้ว่าจำนวนไฟล์อินพุตและจำนวนไฟล์เอาต์พุตควรเท่ากัน เพื่อไม่ให้มีข้อมูลที่หายไป ซึ่งหากมีไฟล์ใดหายไประหว่างดำเนินการ เป็นการยากที่จะหาว่าไฟล์ใดหายไปหากมีไฟล์จำนวนมาก จึงได้ทำการพัฒนาโปรแกรมนี้ขึ้น และพัฒนาด้วย Shell Script เนื่องจาก Shell Script เป็นภาษาที่พัฒนาง่ายและมีความฉลาดในตัวเองค่อนข้างสูง

3.3.7 การแยกข้อความในไฟล์โดยภาษา

โปรแกรมแยกข้อความในไฟล์โดยภาษา ถูกพัฒนาขึ้นเนื่องจากบางครั้งเมื่อบริษัทรับไฟล์ข้อมูลมาจากลูกค้าหรือดึงข้อมูลมาจากอินเทอร์เน็ตจะพบว่าหลายครั้งเราจะได้มาทั้งภาษาอังกฤษและคู่ภาษาในไฟล์เดียวกัน ซึ่งไม่สามารถนำไปประมวลผลต่อได้ ซึ่งมีความจำเป็นที่จะต้องแยกไฟล์ต้นฉบับ 1 ไฟล์ออกมาเป็นไฟล์คู่ภาษา 2 ไฟล์ก่อน เพื่อนำไปแยกประมวลผล จึงได้ทำการพัฒนาโปรแกรมนี้ขึ้น และพัฒนาด้วย Shell Script เนื่องจาก Shell Script เป็นภาษาที่พัฒนาง่าย

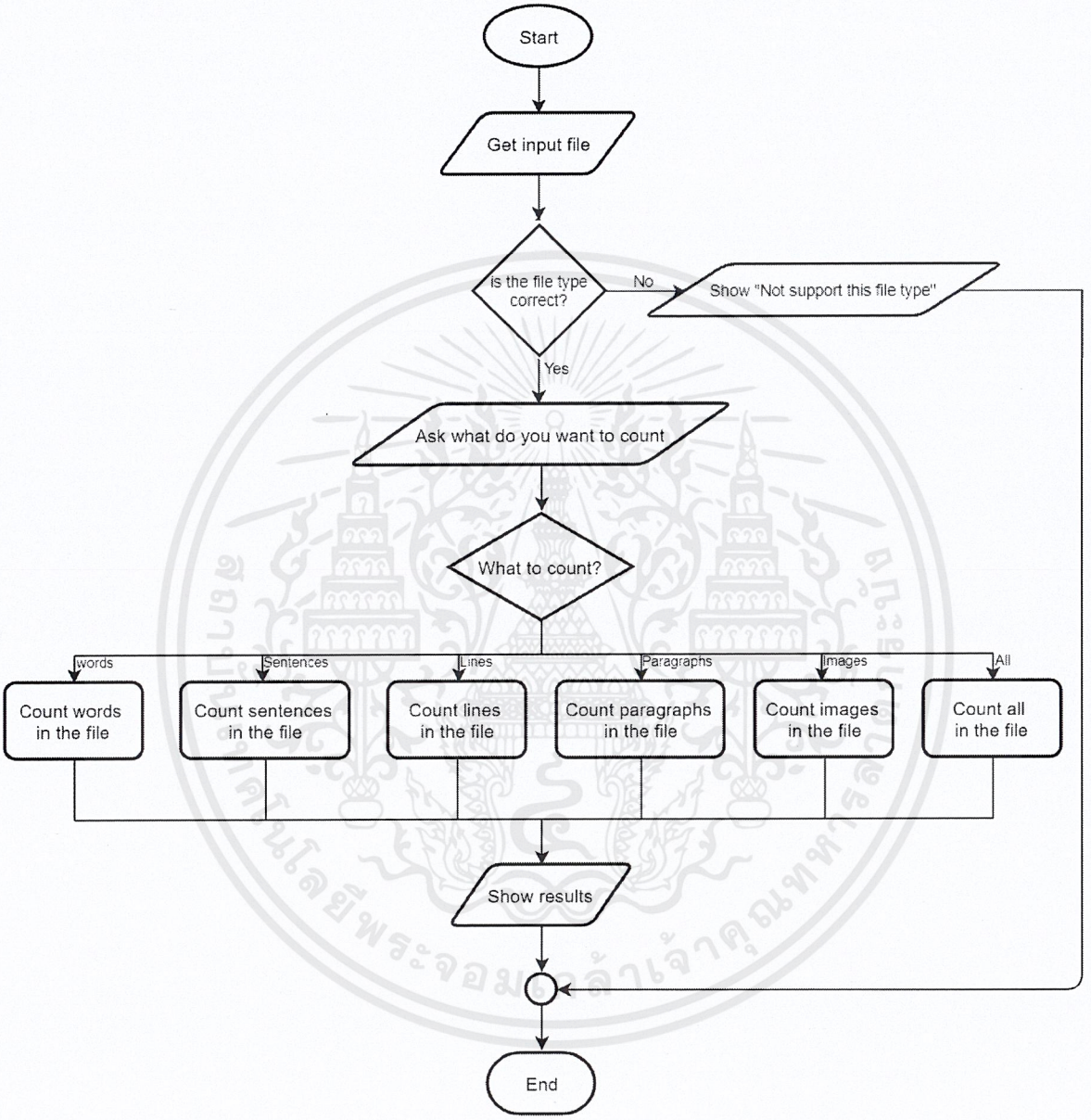
3.3.8 การตรวจสอบและลบข้อความภาษาจีนแบบดั้งเดิม (Traditional) ออกจากไฟล์

โปรแกรมตรวจสอบและลบข้อความภาษาจีนแบบดั้งเดิมออกจากไฟล์ ถูกพัฒนาขึ้นเนื่องจากบางครั้งเมื่อบริษัทรับไฟล์ข้อมูลมาจากลูกค้าหรือดึงข้อมูลมาจากอินเทอร์เน็ตเป็นภาษาจีน จะพบว่าหลายครั้งเราจะได้มาทั้งภาษาจีนแบบดั้งเดิมและภาษาจีนแบบย่อ แต่เนื่องจากบริษัทเน้นผลิตภัณฑ์การแปลภาษา แน่นอนว่าปัจจุบันภาษาจีนแบบย่อได้รับความนิยมเป็นอย่างมาก จึงต้องการใช้ข้อมูลเพียงภาษาจีนแบบย่อเท่านั้น จึงมีความจำเป็นต้องลบภาษาจีนแบบดั้งเดิมออกจากไฟล์ต้นฉบับก่อน จึงได้ทำการพัฒนาโปรแกรมตรวจสอบและลบข้อความภาษาจีนแบบดั้งเดิมออกจากไฟล์ขึ้น โดยพัฒนาด้วยภาษาจาวาเนื่องจากบริษัทรองรับการทำงานโดยภาษาจาวา และ Shell Script เนื่องจากเป็นภาษาที่พัฒนาง่ายและมีความฉลาดในตัวเองค่อนข้างสูง

3.4 แผนภาพอธิบายโครงสร้างและการทำงานของโปรแกรม

3.4.1 การวิเคราะห์โปรแกรมและแผนผังกระบวนการทำงาน (Flowchart)

- การนับคำ ประโยค บรรทัด ย่อหน้าและรูปภาพในไฟล์



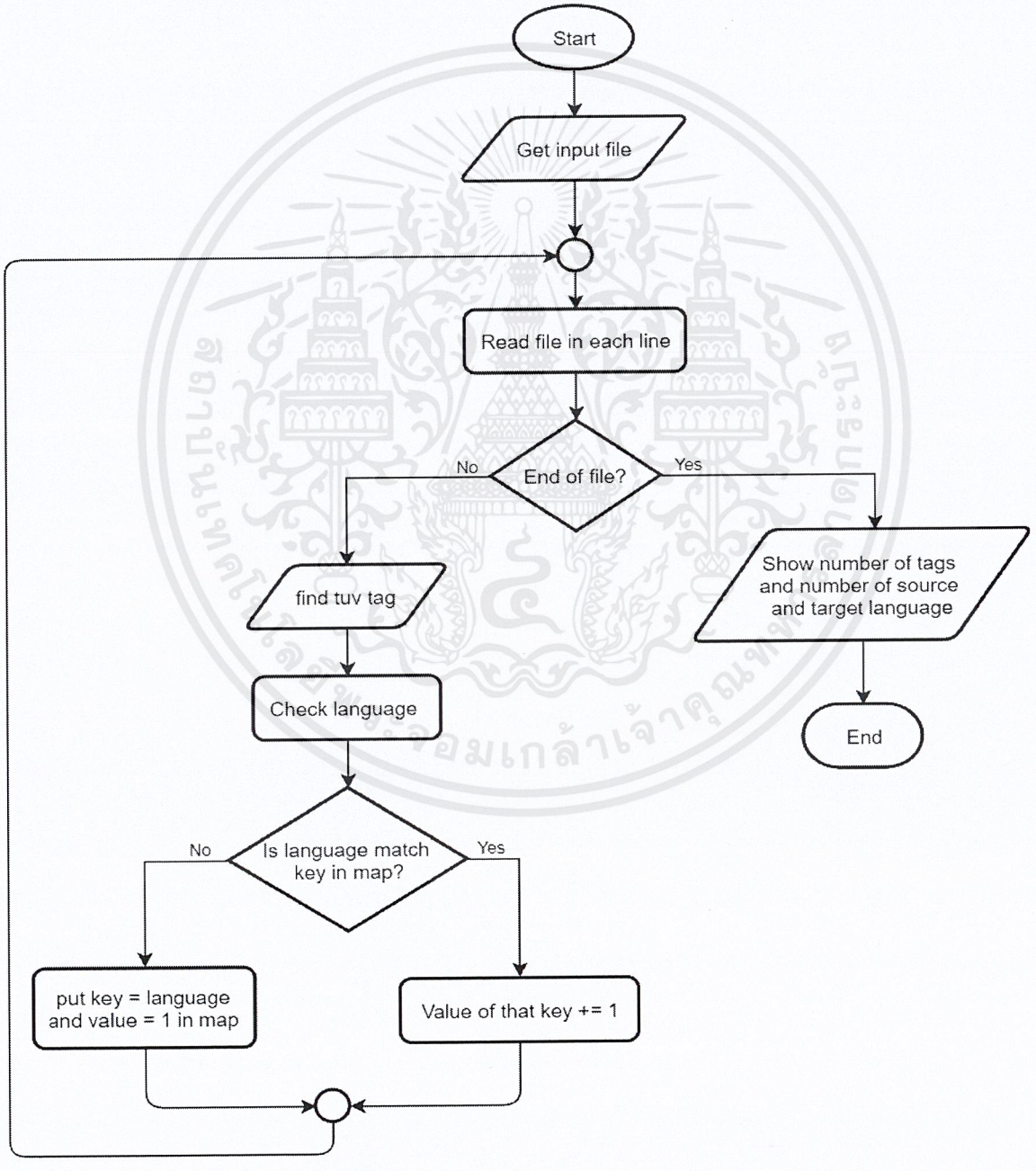
ภาพที่ 3.7 แผนผังการทำงานของโปรแกรมนับคำ ประโยค บรรทัด ย่อหน้าและรูปภาพในไฟล์

โปรแกรมนี้พัฒนาโดยภาษาจาวา ซึ่งการทำงานของโปรแกรมเริ่มจากการรับไฟล์อินพุตเข้ามา จากนั้นทำการตรวจสอบประเภทของไฟล์ว่าตรงกับที่ตัวโปรแกรมรองรับหรือไม่ ซึ่งโปรแกรมนี้รองรับ 5 ประเภทของไฟล์ตามที่กล่าวไปข้างต้น จากนั้นจะทำการแสดงข้อความเพื่อรับอินพุตว่าต้องการนับคำ ประโยค บรรทัด ย่อหน้า รูปภาพหรือนับทั้ง 5 อย่างในไฟล์ เมื่อเลือกสิ่งที่ต้องการนับเรียบร้อยแล้ว

โปรแกรมจะทำการคำนวณและแสดงผลลัพธ์ออกมา แต่หากประเภทของไฟล์ไม่ตรงกับที่โปรแกรมรองรับ โปรแกรมจะแสดงข้อความว่า “Not support this file type”

ประโยชน์ของโปรแกรมนี้ทำให้เราสามารถคำนวณคำ ประโยค บรรทัด ย่อหน้า รูปภาพ ได้อย่างง่ายดาย เพราะการทำงานกับข้อมูลจำนวนมาก ไฟล์ทุกอย่างจะมีขนาดค่อนข้างใหญ่ ซึ่งเราไม่สามารถนับเองได้ จึงจำเป็นต้องมีตัวช่วยในการนับ นอกจากนี้ตัวโปรแกรมยังเป็นส่วนหนึ่งของ LSscript ซึ่งเป็นผลิตภัณฑ์ที่ขายให้กับลูกค้าอีกด้วย

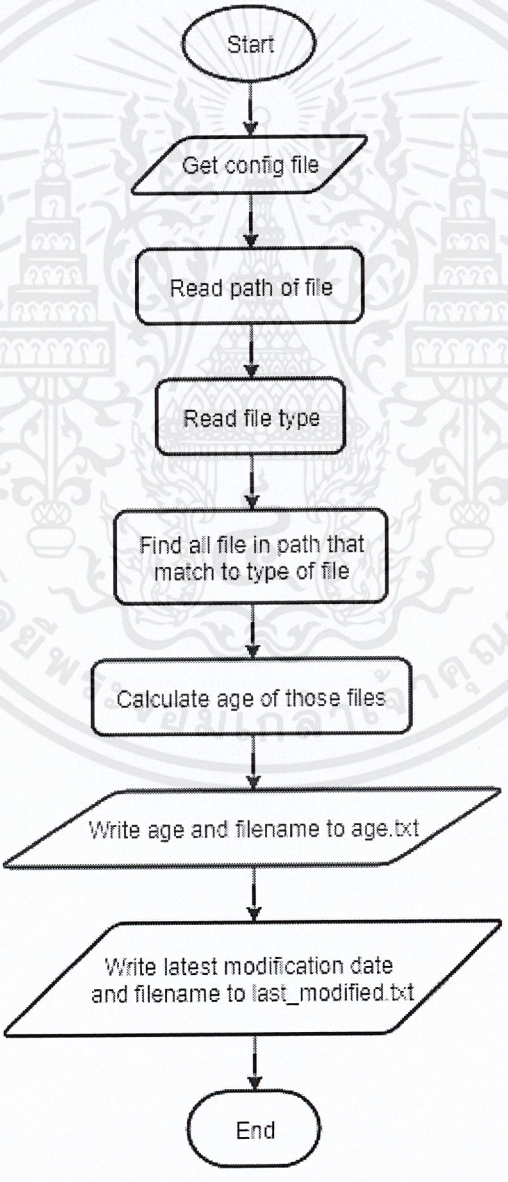
- การตรวจสอบจำนวนคู่ภาษา



ภาพที่ 3.8 แผนผังการทำงานของโปรแกรมตรวจสอบจำนวนคู่ภาษา

โปรแกรมนี้พัฒนาโดยภาษาจาวา ซึ่งการทำงานของโปรแกรมเริ่มจากรับไฟล์อินพุตประเภท XML จากนั้นจะอ่านไฟล์ทีละบรรทัดเพื่อหา Tag tuv เมื่อเจอแล้วจะทำการอ่านว่าใน Tag นั้น มีค่า Lang เป็นภาษาใด และนับจำนวนของ Lang ที่พบในไฟล์ ซึ่งไฟล์หนึ่งควรมี 2 ภาษา (Lang) ที่เป็นคู่ภาษากัน เมื่ออ่านจนจบไฟล์จะตรวจสอบว่าคู่ภาษาทั้งสองฝั่งภาษามีจำนวนเท่ากันหรือไม่ และทำการแสดงผลลัพธ์ชื่อไฟล์นั้นๆคืออะไร มีคู่ภาษาคืออะไร แต่ละภาษามีจำนวนเท่าไหร่ ผลรวมของคู่ภาษาที่สามารถนำไปใช้งานมีจำนวนเท่าไหร่ โดยดูจากจำนวนของทั้ง 2 ภาษาว่ามีเท่ากันหรือไม่ ถ้าเท่ากัน จะนำจำนวนของทั้งสองภาษามาบวกกัน แต่หากไม่เท่ากันจะนำจำนวนของภาษาที่มีน้อยกว่ามาคูณสอง และสุดท้ายจะแสดงผลรวมของคู่ภาษาที่สามารถนำไปใช้งานของทุกๆไฟล์

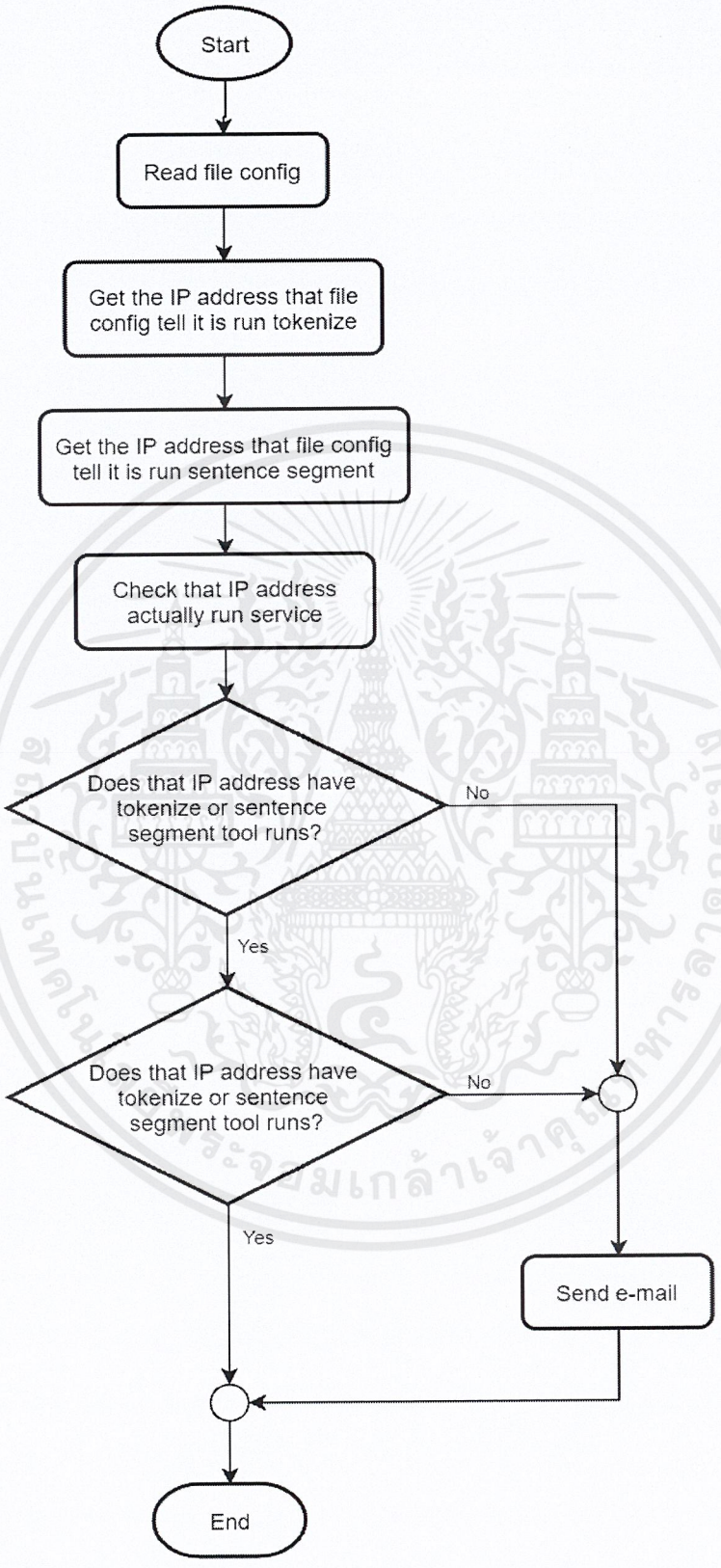
- การตรวจสอบอายุของไฟล์และวันที่มีการปรับปรุงไฟล์ล่าสุด



ภาพที่ 3.9 แผนผังการทำงานของโปรแกรมตรวจสอบอายุของไฟล์และวันที่มีการปรับปรุงไฟล์ล่าสุด

โปรแกรมนี้พัฒนาโดย Shell Script ซึ่งการทำงานของโปรแกรมเริ่มจากอ่าน Config File ทีละบรรทัดว่าต้องการตรวจสอบไฟล์จากไดเรกทอรีใดและเป็นไฟล์ประเภทใด จากนั้นโปรแกรมจะทำการดึงไฟล์ที่ตรงตามที่ Config File ออกมาแล้วคำนวณว่าไฟล์นั้นๆมีอายุกี่วันนับจากวันที่ทำการแก้ไขไฟล์ล่าสุดและบันทึกผลลัพธ์ที่ได้ลงไฟล์ age.txt นอกจากนี้ยังตรวจสอบว่าไฟล์นั้นมีการแก้ไขล่าสุดเมื่อไรและบันทึกผลลัพธ์ที่ได้ลงไฟล์ last_modified.txt

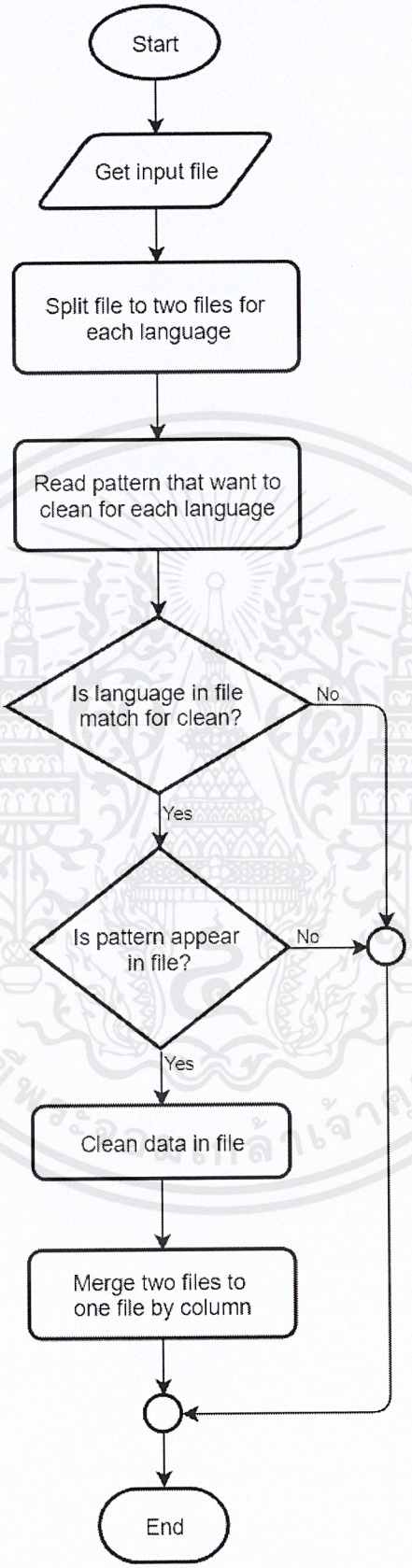
- การตรวจสอบการทำงานของเครื่อง Server ที่ใช้ในการทำงานในระบบ



ภาพที่ 3.10 แผนผังการทำงานของโปรแกรมตรวจสอบการทำงานของเครื่องเซิร์ฟเวอร์ที่ใช้ในการทำงานในระบบ

จากภาพที่ 3.10 จะเห็นว่าเป็นแผนผังการทำงานของโปรแกรมตรวจการทำงานของเครื่องเซิร์ฟเวอร์ที่ใช้ในการทำงานในระบบโดยภาพรวม ซึ่งในรายละเอียดจริงนั้น โปรแกรมนี้จะทำงานบนเครื่องเซิร์ฟเวอร์จำนวน 2 เครื่องที่เป็นเครื่องแม่ที่อยู่ในประเทศไทยและประเทศสิงคโปร์ โดยทำงานเหมือนกัน คือเริ่มจากการอ่าน Config File ที่มีอยู่แล้วในเครื่อง ซึ่งใน Config File จะบอกหมายเลขไอพีของเครื่องเซิร์ฟเวอร์อื่นที่อยู่ในวงเดียวกันว่ามีเครื่องใดบ้างที่ทำงานอยู่และเครื่องใดบ้างควรมีการทำ Sentence Segment หรือ Tokenize อยู่ จากนั้นจะเข้าไปทำการตรวจสอบการทำงานของแต่ละไอพีว่ามีโปรแกรมทำงานนั้นๆอยู่จริงหรือไม่ หากจริงก็ทำการตรวจสอบว่ามีเซอร์วิสทำงานอยู่จริงหรือไม่ เนื่องจากเคยเกิดเหตุการณ์ที่ตรวจสอบแล้วพบว่าตัวโปรแกรมทำงานอยู่แต่เซอร์วิสไม่ทำงาน จึงจำเป็นต้องตรวจสอบทั้ง 2 กรณี โดยเครื่องทั้งสองจะทำการบันทึกลงไฟล์ summarys4u.log สำหรับเครื่องที่ประเทศสิงคโปร์และ summarybkk.log สำหรับเครื่องที่ประเทศไทย ว่าแต่ละไอพีควรมีการทำ Sentence Segment หรือ Tokenize อยู่ และไอพีนั้นทำอยู่จริงหรือไม่ (True/False) จากนั้นเครื่องที่ประเทศไทยจะทำการดึงไฟล์ summarys4u.log จากเครื่องที่ประเทศสิงคโปร์มารวมกับไฟล์ summarybkk.log เป็นไฟล์สรุปไฟล์เดียว ว่ามีไอพีใดที่ตัวโปรแกรมไม่ทำงาน หรือเซอร์วิสไม่ทำงานหรือทั้งสองกรณีหรือไม่ หากมี จะทำการส่งอีเมลเพื่อแจ้งว่ามีปัญหาที่ไอพีใด และอยู่ในขั้นตอน Sentence segment หรือ Tokenize ซึ่งในขั้นตอนของการตรวจสอบว่ามีเซอร์วิสทำงานอยู่จริงหรือไม่ ใช้ NodeJS ในการพัฒนา ส่วนในขั้นตอนอื่นๆใช้ Shell Script

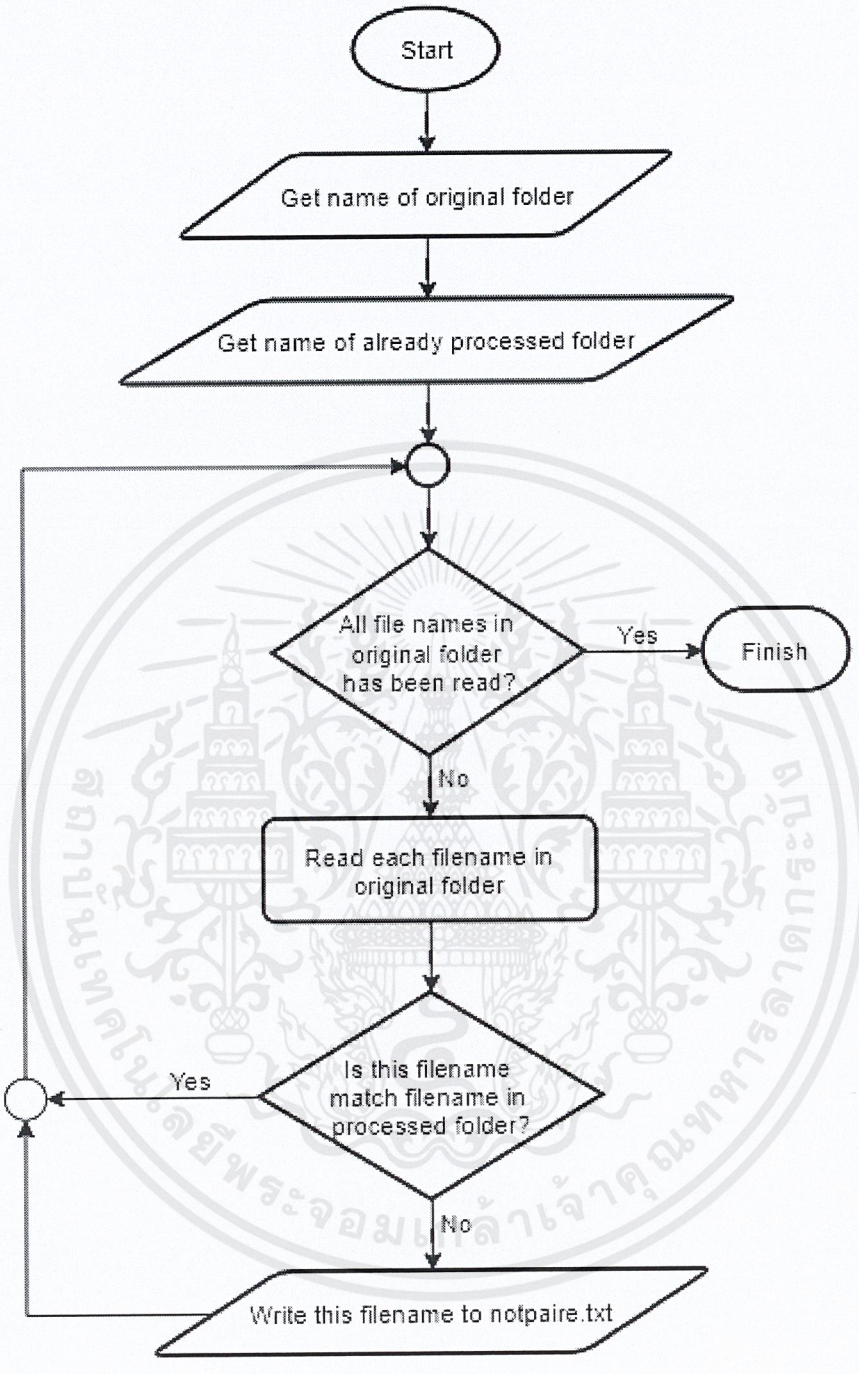
- การทำความสะอาดข้อมูลก่อนนำไปประมวลผล



ภาพที่ 3.11 แผนผังการทำงานของโปรแกรมทำความสะอาดข้อมูลก่อนนำไปประมวลผล

โปรแกรมนี้พัฒนาโดย Shell Script โดยการทำงานของโปรแกรมเริ่มจากรับไฟล์อินพุตที่มีสองคอลัมน์ ซึ่งเป็นคู่ภาษากันที่คั่นด้วยเครื่องหมาย Tab โดยมีเงื่อนไขในการตั้งชื่อไฟล์อินพุต คือ ต้องเป็นตัวย่อของภาษาแรก คั่นด้วย _ และตามด้วยตัวย่อของภาษาที่สอง เช่น EN_CS.txt จากนั้นจะทำการแยกคู่ภาษาเป็นภาษาละ 1 ไฟล์โดยแยกจากคอลัมน์ในไฟล์อินพุต จะได้ไฟล์ 2 ไฟล์ โดยมีชื่อของไฟล์ ดังนี้ EN_CS_EN.txt หมายถึง ไฟล์นี้เป็นไฟล์ที่แยกออกมาจากไฟล์ EN_CS.txt ซึ่งไฟล์นี้ได้แยกภาษาอังกฤษ (EN) ออกมา และ EN_CS_CS.txt หมายถึง ไฟล์นี้เป็นไฟล์ที่แยกออกมาจากไฟล์ EN_CS.txt ซึ่งไฟล์นี้ได้แยกภาษาเช็ก (CS) ออกมา จากนั้นจะทำการอ่าน Config File ที่ระบุว่าต้องการลบรูปแบบใดในภาษาใด จากนั้นในแต่ละไฟล์จะทำการตรวจสอบว่าเป็นภาษาที่ต้องการทำความสะอาดที่อยู่ใน Config File หรือไม่ หากเป็นภาษาที่ตรงกันจะตรวจสอบต่อว่ามีรูปแบบที่ต้องการลบหรือไม่ หากมีจะทำการทำความสะอาดรูปแบบนั้น เมื่อทำสำเร็จทั้งสองไฟล์ จะทำการรวมทั้งสองไฟล์คู่ภาษาเป็น 1 ไฟล์และใช้ชื่อดังเดิม

- การตรวจสอบและหาชื่อไฟล์ที่ยังไม่ถูกประมวลผล

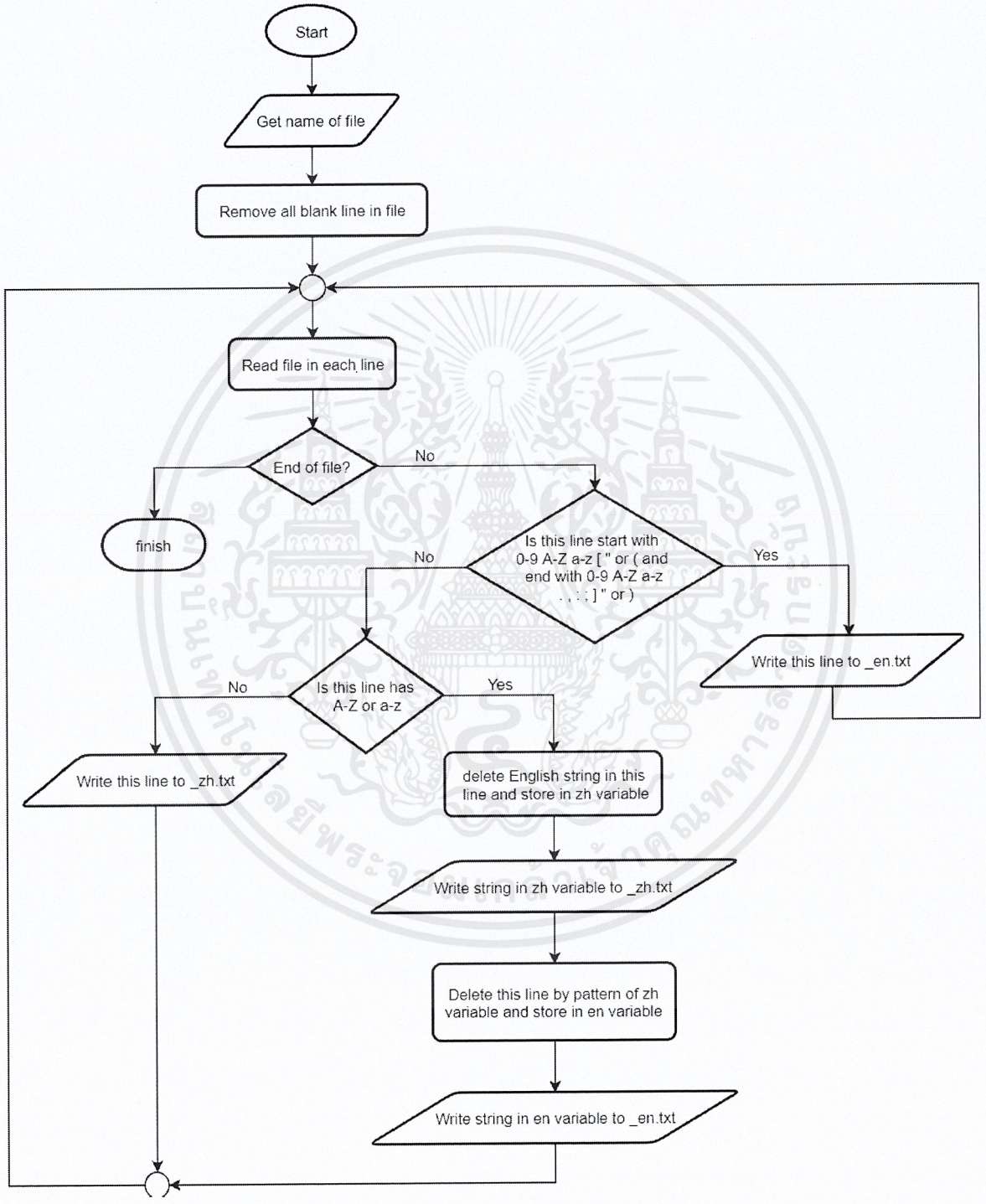


ภาพที่ 3.12 แผนผังการทำงานของโปรแกรมตรวจสอบและหาชื่อไฟล์ที่ยังไม่ถูกประมวลผล

โปรแกรมนี้พัฒนาโดย Shell Script โดยการทำงานของโปรแกรมจะตรวจสอบชื่อไฟล์จากโฟลเดอร์อินพุตและโฟลเดอร์เอาต์พุตเมื่อผ่านการประมวลผลบางอย่าง ซึ่งชื่อไฟล์หลังถูกประมวลผลจะไม่เปลี่ยนไปจากเดิมมาก (ต้องขึ้นต้นชื่อไฟล์ด้วยชื่อลูกค้าเสมอ) ดังนั้น โปรแกรมจะทำการอ่านชื่อไฟล์อินพุตทีละไฟล์และตรวจสอบว่าในโฟลเดอร์เอาต์พุตมีชื่อไฟล์ที่ขึ้นต้นด้วยชื่อลูกค้าเดียวกันกับไฟล์อินพุตหรือไม่

หากไม่มีชื่อลูกค้าที่ตรงกันจะทำการแสดงผลชื่อไฟล์นั้นออกมาในไฟล์เอาต์พุตชื่อว่า notpair.txt เพื่อให้ทีมทำการประมวลผลไฟล์นั้นใหม่อีกครั้ง

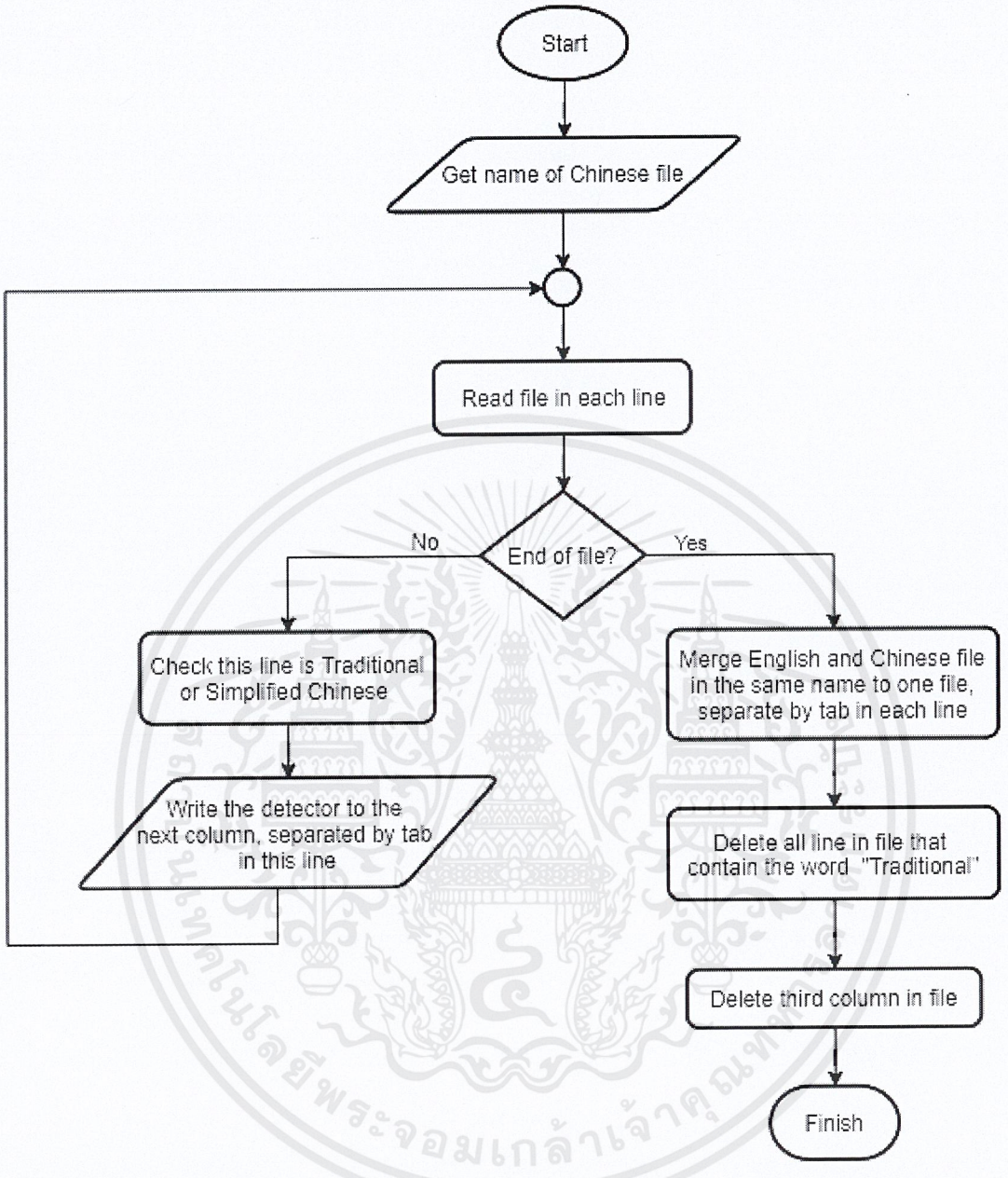
- การแยกข้อความในไฟล์โดยภาษา



ภาพที่ 3.13 แผนผังการทำงานของโปรแกรมแยกข้อความในไฟล์โดยภาษา

โปรแกรมนี้พัฒนาโดย Shell Script ซึ่งการทำงานของโปรแกรมจะทำการแยกไฟล์ 1 ไฟล์ ซึ่งมีทั้งข้อความภาษาจีนและภาษาอังกฤษออกเป็น 2 ไฟล์ คือ ไฟล์สำหรับภาษาจีนและไฟล์สำหรับภาษาอังกฤษ โดยเริ่มจากการลบบรรทัดว่างออกจากไฟล์ต้นฉบับก่อน จากนั้นอ่านไฟล์ทีละบรรทัดโดยตรวจสอบแต่ละบรรทัดว่าขึ้นต้นด้วยตัวเลข (0-9) ตัวอักษรภาษาอังกฤษ (A-Z หรือ a-z) หรือ [“ (และลงท้ายด้วยตัวเลข (0-9) ตัวอักษรภาษาอังกฤษ (A-Z หรือ a-z) หรือ . , : ;] ”) หรือไม่ หากเป็นจริงแสดงว่าบรรทัดนั้นเป็นบรรทัดที่มีแต่ภาษาอังกฤษ จะทำการเขียนบรรทัดนั้นลงไฟล์ _en.txt แต่หากไม่จริงจะทำการตรวจสอบว่าในบรรทัดนั้นๆมีตัวอักษรภาษาอังกฤษ (A-Z หรือ a-z) อยู่หรือไม่ หากเป็นจริงแสดงว่าบรรทัดนั้นๆมีทั้งข้อความส่วนภาษาอังกฤษและภาษาจีน จะทำการลบส่วนของภาษาอังกฤษออกให้เหลือเฉพาะส่วนของภาษาจีน เก็บข้อความนั้นๆลงตัวแปรหนึ่ง (ในที่นี่ใช้ชื่อตัวแปรว่า zh) จากนั้นเขียนข้อความที่อยู่ในตัวแปร zh ลงไฟล์ _zh.txt ในขณะเดียวกันก็จะทำการลบข้อความบรรทัดเดิมด้วยรูปแบบของ zh จะเหลือเพียงส่วนที่เป็นข้อความภาษาอังกฤษ เก็บข้อความนั้นๆลงตัวแปรหนึ่ง (ในที่นี่ใช้ชื่อตัวแปรว่า en) จากนั้นเขียนข้อความที่อยู่ในตัวแปร en ลงไฟล์ _en.txt แต่หากไม่เป็นจริงแสดงว่าบรรทัดนั้นเป็นบรรทัดที่มีแต่ภาษาจีน จะทำการเขียนบรรทัดนั้นลงไฟล์ _zh.txt ทำเช่นนี้ไปเรื่อยๆจนถึงบรรทัดสุดท้ายของไฟล์ต้นฉบับ

- การตรวจสอบและลบข้อความภาษาจีนแบบดั้งเดิม (Traditional) ออกจากไฟล์



ภาพที่ 3.14 แผนผังการทำงานของโปรแกรมตรวจสอบและลบข้อความภาษาจีนแบบดั้งเดิมออกจากไฟล์

โปรแกรมนี้พัฒนาโดย Shell Script และภาษาจาวา โดยการทำงานของโปรแกรมจะทำการตรวจสอบข้อความภาษาจีนว่าเป็นภาษาจีนแบบดั้งเดิม (Traditional) หรือภาษาจีนแบบย่อ (Simplified) โดยเริ่มจากรับข้อความภาษาจีนทีละบรรทัดจากไฟล์ ตัวอย่างเช่น 医生。 จากนั้นจะเขียนสิ่งที่ตรวจสอบได้ต่อจากข้อความภาษาจีนในบรรทัดนั้นๆ โดยคั่นด้วย Tab (\t) ตัวอย่างเช่น 医生。 Simplified และทำเช่นนี้ต่อไปเรื่อยๆจนถึงบรรทัดสุดท้ายของไฟล์ เมื่อสิ้นสุดไฟล์จะทำการรวม 2 ไฟล์ที่ชื่อลูกค้าเดียวกันและเนื้อความในไฟล์มีความหมายเดียวกัน ต่างกันที่ภาษาเข้าด้วยกัน คือ ไฟล์ฝั่ง

ภาษาอังกฤษและไฟล์ฝั่งภาษาจีน โดยข้อความฝั่งภาษาอังกฤษจะอยู่ในคอลัมน์ที่ 1 ข้อความฝั่งภาษาจีนจะอยู่ในคอลัมน์ที่ 2 และสิ่งที่ตรวจสอบได้จะอยู่ในคอลัมน์ที่ 3 จากนั้นทำการลบบรรทัดที่มีคำว่า Traditional ออกจากไฟล์ จากนั้นลบคอลัมน์ที่ 3 ออกจากไฟล์ทั้งหมด เพื่อให้เหลือเพียงคู่ภาษาจีนและอังกฤษเท่านั้น โดยในส่วนของการตรวจสอบภาษาจีน จะใช้ Tool ที่พัฒนาโดยภาษาจาวา ส่วนขั้นตอนอื่นๆพัฒนาโดยใช้ Shell Script

บทที่ 4
ผลการวิจัย

ในบทนี้จะกล่าวถึงผลลัพธ์ที่ได้จากการทำงานของระบบแต่ละระบบว่ามีผลลัพธ์ออกมาเป็นอย่างไร เป็นที่น่าพอใจหรือไม่ เพราะเหตุใด โดยจะแยกผลการทำงานเป็น 8 ส่วนตามโปรแกรมต่างๆที่ได้พัฒนา

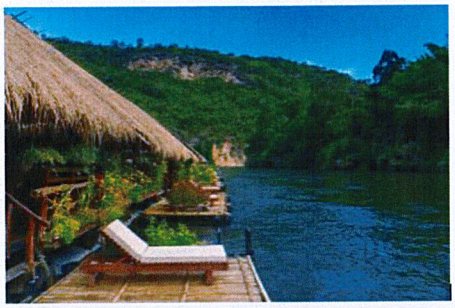
4.1 การนับคำ ประโยค บรรทัด ย่อหน้าและรูปภาพในไฟล์

โปรแกรมนี้เป็นโปรแกรมสำหรับนับคำ ประโยค บรรทัด ย่อหน้าและจำนวนรูปภาพในไฟล์ 5 ประเภท คือ Document (.doc และ .docx), Power point (.ppt และ .pptx), Excel (.xls และ .xlsx), Text และ Portable Document Format (PDF) โดยไฟล์แต่ละประเภทสามารถเลือกนับสิ่งต่างๆ ได้ดังนี้

ตารางที่ 4.1 แสดงความสัมพันธ์ระหว่างประเภทไฟล์และสิ่งที่สามารถนับได้

ประเภทไฟล์	นับคำ	นับประโยค	นับบรรทัด	นับย่อหน้า	นับรูปภาพ
Document	√	√	-	√	√
Power point	√	√	√	√	√
Excel	√	√	√	-	√
Text	√	√	√	√	-
PDF	√	√	√	√	√

ในโปรแกรมนี้ผู้จัดทำได้ทดสอบกับไฟล์ทั้ง 5 ประเภท (8 นามสกุล) พบว่า เวลาที่ใช้ขึ้นอยู่กับขนาดของไฟล์ในลักษณะแปรผันตรง แต่ระยะเวลาไม่ถึงกับนานมาก ซึ่งอยู่ในระดับที่น่าพอใจและสามารถนำมาใช้จริงได้ โดยผู้จัดทำได้ยกตัวอย่างมา 2 ไฟล์ คือ ไฟล์ PDF (.pdf) ดังภาพที่ 4.1, 4.2 และ 4.3 และไฟล์ Excel (.xlsx) ดังภาพที่ 4.4 และ 4.5



After the busy buzz of Thailand's capital, the pleasant vibe of Mae Nam Kwae Road along the River Kwai is just what a traveler needs. The stretch that parallels the river is crammed with guesthouses, cafes and bars for eating and socializing.

Although the road isn't overly relaxing, serenity can be found just behind it. Many of the cafes and guesthouses have green gardens with lounge areas that back up to the river. Enjoy a lazy afternoon in a hammock beneath a plumeria tree or on a deck with a cold Chang,Leo or Singha in hand. But try not to lose your Zen when the occasional party boat passes by blaring full-volume karaoke or disco.

ภาพที่ 4.1 ตัวอย่างไฟล์ PDF (.pdf)

```
--- exec-maven-plugin:1.2.1:exec (default-cli) @ count ---
choose number 1:word 2:sentence 3:line 4:paragraph 5:image 6:all -->
6

river_kwai.pdf
Word            = 116
sentense        = 5
line            = 8
paragraph       = 2
Image           = 1
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time: 3.814s
Finished at: Fri Dec 13 21:52:32 ICT 2019
Final Memory: 5M/121M
```

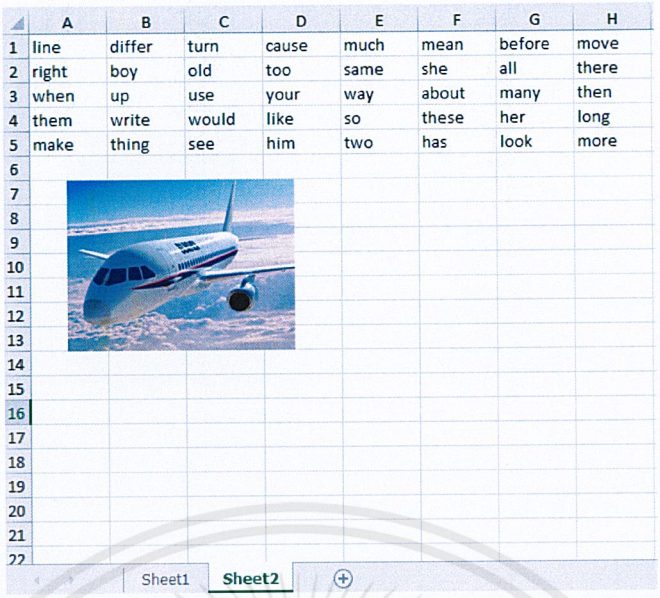
ภาพที่ 4.2 ผลลัพธ์ที่ได้จากโปรแกรมนับคำ ประโยค บรรทัด ย่อหน้าและรูปภาพในไฟล์ PDF (.pdf)

	A	B	C	D	E	F	G	H
1	as	I	his	that	he	was	for	on
2	are	with	they	be	at	one	have	this
3	from	by	hot	word	but	what	some	is
4	it	you	or	had	the	of	to	and
5	a	in	we	can	out	other	were	which
6	do	their	time	if	will	how	said	an
7	each	tell	does	set	three	want	air	well
8	also	play	small	end	put	home	read	hand
9	port	large	spell	add	even	land	here	must
10								
11	Excuse me, could you tell me the way to the station, please?							
12	Excuse me, I'm looking for the town hall.							
13	How far is it from the church to the station?							
14	Is it far from the church to the station?							
15	It takes about 10 minutes by bus.							
16	It's a 10-minute walk.							
17	The church is within walking distance.							
18								
19								
20								
21								
22								

(Ctrl)

Sheet1 Sheet2

ภาพที่ 4.3 ตัวอย่างไฟล์ Excel (.xlsx) Sheet ที่ 1



ภาพที่ 4.4 ตัวอย่างไฟล์ Excel (.xlsx) Sheet ที่ 2

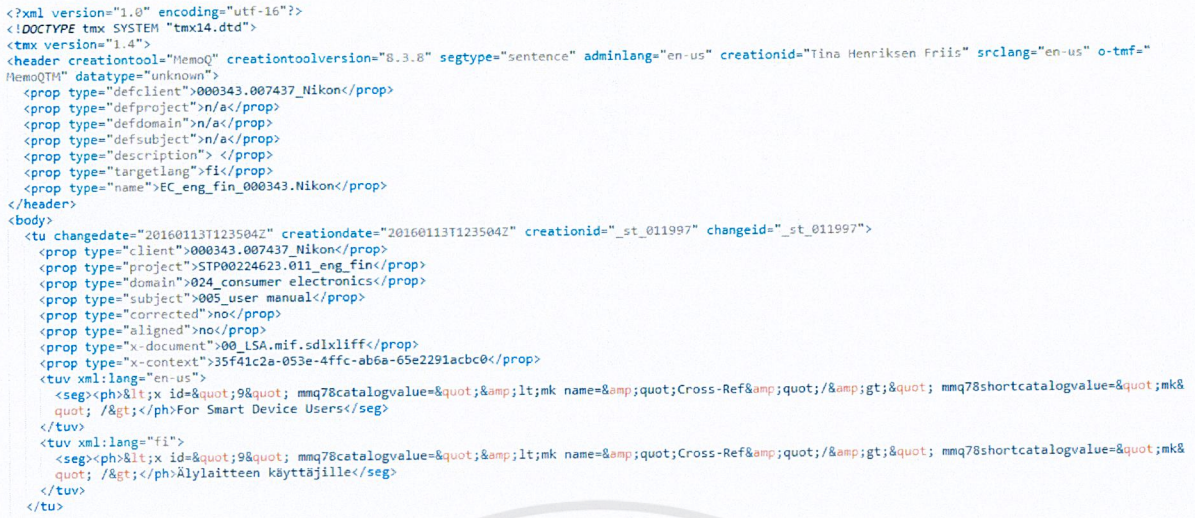
```
--- exec-maven-plugin:1.2.1:exec (default-cli) @ count ---
choose number 1:word 2:sentence 3:line 4:image 5:all -->
5

glossary.xlsx
word            = 168
sentense        = 7
line            = 21
Image           = 1
------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------
Total time: 3.572s
Finished at: Fri Dec 13 19:23:05 ICT 2019
Final Memory: 6M/153M
```

ภาพที่ 4.5 ผลลัพธ์ที่ได้จากโปรแกรมนับคำ ประโยค บรรทัด ย่อหน้าและรูปภาพในไฟล์ Excel (.xlsx)

จากการทดสอบโปรแกรม พบว่า โปรแกรมสามารถประมวลผลได้อย่างรวดเร็ว และมีความถูกต้องอยู่ที่ 99% เนื่องจากภาษาคอมพิวเตอร์ที่เขียนเพื่อนับประโยค เขียนโดยนักพัฒนาในทีมโดยใช้วิธีการ Regular Expression จึงมีโอกาสที่บางประโยคจะหลุดเงื่อนไขของภาษาคอมพิวเตอร์ไปได้ แต่เป็นส่วนน้อยเท่านั้น ซึ่งอยู่ในความพึงพอใจที่รับได้

4.2 การตรวจสอบจำนวนคู่ภาษา

โปรแกรมนี้เป็นโปรแกรมสำหรับนับจำนวนคู่ของคู่ภาษาในไฟล์ XML ของลูกค้า เพื่อเป็นการตรวจสอบในขั้นแรกว่าแต่ละข้อความมีทั้งภาษาอังกฤษและคู่ภาษาที่มีความหมายเดียวกันครบคู่



ภาพที่ 4.6 ตัวอย่างไฟล์ XML ที่ต้องตรวจสอบจำนวนคู่ภาษาใน Tag tuv

ในโปรแกรมนี้ผู้จัดทำได้ทดสอบกับไฟล์ต้นฉบับจริงจากลูกค้าจำนวน 6 ไฟล์ แต่ละไฟล์มีชื่อ จำนวนบรรทัดและขนาดไฟล์ ดังนี้

ตารางที่ 4.2 แสดงความสัมพันธ์ระหว่างชื่อ จำนวนบรรทัดและขนาดของไฟล์ที่นำมาใช้ทดสอบโปรแกรมตรวจสอบจำนวนคู่ภาษา

ชื่อไฟล์	จำนวนบรรทัด	ขนาดไฟล์ (MB)
de-sl_GNOME.tmx	18	1
EC_eng_fin_000343.Nikon.tmx	199,768	15
en-ko.tmx	11,308	2
PP-ENKO2.tmx	586,853	17
R_DT_eng_fin_mechanical_engineering.tmx	5,884,718	298
TMX_Test.tmx	2,027,641	115

จากตารางที่ 4.2 จะเห็นว่าไฟล์ที่นำมาทดสอบกับโปรแกรมมีจำนวน 6 ไฟล์ ซึ่งมีบรรทัดของจำนวนไฟล์ทั้ง 6 ไฟล์รวมกันอยู่ที่ 8,710,306 บรรทัด เมื่อทดสอบกับโปรแกรม ได้ผลลัพธ์ ดังนี้

```
de-sl_GNOME.tmx          6
   language: {de=3, sl=3}

EC_eng_fin_000343.Nikon.tmx              24962
   language: {en-us=12483, fi=12481}

en-ko.tmx                5652
   language: {en-US=2826, ko-KR=2826}

PP-ENKO2.tmx             117502
   language: {EN-US=58751, KO-KR=58751}

R_DT_eng_fin_mechanical_engineering.tmx          782274
   language: {fi=391137, en=391137}

TMX_Test.tmx             503104
   language: {de-DE=251552, en-US=251552}

summary 1433500
BUILD SUCCESSFUL (total time: 9 seconds)
```

ภาพที่ 4.7 ผลลัพธ์ที่ได้จากโปรแกรมตรวจสอบจำนวนคู่ภาษา

จากภาพที่ 4.7 พบว่าไฟล์ทั้ง 6 ไฟล์ถูกประมวลผลโดยใช้เวลา 9 วินาที ซึ่งถือว่าเป็นเวลาที่น่าพอใจเมื่อเทียบกับบรรทัดของจำนวนไฟล์ทั้ง 6 ไฟล์รวมกัน โดยเฉลี่ยประมวลผลอยู่ที่ 967,811 บรรทัดต่อ 1 วินาที

4.3 การตรวจสอบอายุของไฟล์และวันที่มีการปรับปรุงไฟล์ล่าสุด

โปรแกรมนี้เป็นโปรแกรมสำหรับตรวจสอบอายุของไฟล์เฉพาะนามสกุลที่เราต้องการและแสดงวันที่มีการปรับปรุงของไฟล์ล่าสุด

ในโปรแกรมนี้ผู้จัดทำได้นำมาใช้ในการทำงานจริงเพื่อหาไฟล์นามสกุลใดใน Path ที่ต้องการ แต่ไม่ทราบแน่ชัดว่าเป็น Path ใด โดยผลลัพธ์ที่ได้จากโปรแกรม จะได้ไฟล์ 2 ไฟล์ คือ age.txt ที่แสดงอายุของไฟล์ ดังภาพที่ 4.8 และ last_modified.txt ที่แสดงวันที่มีการปรับปรุงไฟล์ล่าสุด ดังภาพที่ 4.9

```
thapanee@DESKTOP-TR7DVSC:/mnt/d/work2.2561/project_coop/script/check_age_and_lastmodified/check_age_and_lastmodified$ bash check_age_and_lastmodified.sh file.config
thapanee@DESKTOP-TR7DVSC:/mnt/d/work2.2561/project_coop/script/check_age_and_lastmodified/check_age_and_lastmodified$ head -50 age.txt
156 /home/thapanee/.subversion/Kam/mavenproject1/sample_file_test/sample_file_test/pdf_EnCh.pdf
156 /home/thapanee/.subversion/Kam/mavenproject1/sample_file_test/sample_file_test/pdf_EngOnly.pdf
191 /home/thapanee/.local/lib/python2.7/site-packages/Click-7.0.dist-info/LICENSE.txt
191 /home/thapanee/.local/lib/python2.7/site-packages/Click-7.0.dist-info/top_level.txt
191 /home/thapanee/.local/lib/python2.7/site-packages/PyYAML-5.1.dist-info/LICENSE.txt
191 /home/thapanee/.local/lib/python2.7/site-packages/PyYAML-5.1.dist-info/top_level.txt
191 /home/thapanee/.local/lib/python2.7/site-packages/aspy.yaml-1.3.0.dist-info/top_level.txt
191 /home/thapanee/.local/lib/python2.7/site-packages/certifi-2019.3.9.dist-info/LICENSE.txt
191 /home/thapanee/.local/lib/python2.7/site-packages/certifi-2019.3.9.dist-info/top_level.txt
191 /home/thapanee/.local/lib/python2.7/site-packages/cfgv-2.0.0.dist-info/top_level.txt
```

ภาพที่ 4.8 ผลลัพธ์จากไฟล์ age.txt (แสดงอายุของไฟล์)

```
thapanee@DESKTOP-TR7DVSC:/mnt/d/work2.2561/project_coop/script/check_age_and_lastmodified/check_age_and_lastmodified$ head -10 last_modified.txt
07/09/19        /home/thapanee/.subversion/Kam/mavenproject1/sample_file_test/sample_file_test/pdf_EnCh.pdf
07/09/19        /home/thapanee/.subversion/Kam/mavenproject1/sample_file_test/sample_file_test/pdf_EngOnly.pdf
06/04/19        /home/thapanee/.local/lib/python2.7/site-packages/Click-7.0.dist-info/LICENSE.txt
06/04/19        /home/thapanee/.local/lib/python2.7/site-packages/Click-7.0.dist-info/top_level.txt
06/04/19        /home/thapanee/.local/lib/python2.7/site-packages/PyYAML-5.1.dist-info/LICENSE.txt
06/04/19        /home/thapanee/.local/lib/python2.7/site-packages/PyYAML-5.1.dist-info/top_level.txt
06/04/19        /home/thapanee/.local/lib/python2.7/site-packages/aspy.yaml-1.3.0.dist-info/top_level.txt
06/04/19        /home/thapanee/.local/lib/python2.7/site-packages/certifi-2019.3.9.dist-info/LICENSE.txt
06/04/19        /home/thapanee/.local/lib/python2.7/site-packages/certifi-2019.3.9.dist-info/top_level.txt
06/04/19        /home/thapanee/.local/lib/python2.7/site-packages/cfgv-2.0.0.dist-info/top_level.txt
```

ภาพที่ 4.9 ผลลัพธ์จากไฟล์ last_modified.txt (แสดงวันที่มีการปรับปรุงไฟล์ล่าสุด)

4.4 การตรวจสอบการทำงานของเครื่อง Server ที่ใช้ในการทำงานในระบบ

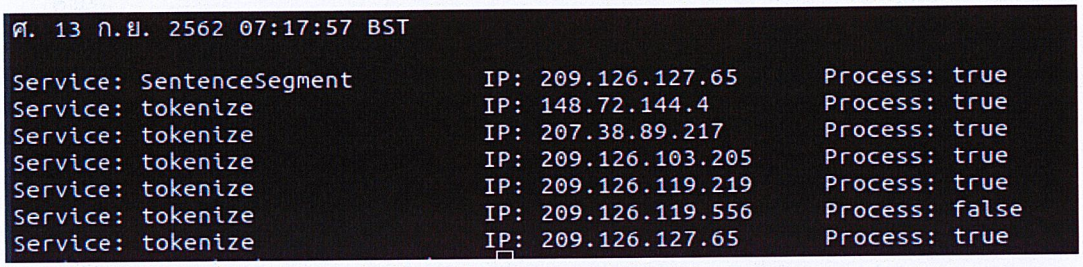
โปรแกรมนี้เป็นโปรแกรมสำหรับตรวจสอบการทำงานของเครื่อง Server ที่ใช้ในการทำงานในระบบทั้งหมดว่ามีการดำเนินการ Sentence Segment และ Tokenize อยู่หรือไม่ หากพบว่ามีเครื่องเซิร์ฟเวอร์ลูกเครื่องใดที่ Tool ในการทำ Sentense Segment/Tokenize ไม่ทำงาน หรือ Tool ทำงานแต่ Service ที่เรียกใช้ไม่ทำงาน จะทำการส่งอีเมลไปที่หัวหน้าโครงการ เพื่อรับทราบและแก้ไขอย่างทันถ่วงที

ในโปรแกรมนี้ผู้จัดทำได้ติดตั้งลงเครื่องเซิร์ฟเวอร์หลักจำนวน 2 เครื่อง คือ เครื่องเซิร์ฟเวอร์ที่ประเทศไทย 1 เครื่อง (ที่บริษัทเรียกว่า เครื่อง Bangkok หรือ bkk) และเครื่องเซิร์ฟเวอร์ที่ประเทศสิงคโปร์ 1 เครื่อง (ที่บริษัทเรียกว่า เครื่อง Server for You หรือ s4u) ซึ่งตัวโปรแกรมจะมีการทำงานทุกๆ 30 นาที เมื่อทดสอบโปรแกรม พบว่าตัวโปรแกรมทำงานได้อย่างถูกต้องและลดระยะเวลาในการเข้าไปตรวจสอบการทำงานของเครื่องเซิร์ฟเวอร์แต่ละตัวได้อย่างมาก เนื่องจากเครื่องเซิร์ฟเวอร์ที่ใช้มีจำนวนหลายเครื่อง หากต้องการตรวจสอบเอง จำเป็นต้อง Secure Shell เข้าแต่ละเครื่องและมีความจำเป็นต้องรู้รหัสผ่านของแต่ละเครื่องอีกด้วย

```
172.17.101.237
tomcat     222 11.4 17.8 13894252 1280228 ?     Sl   ก.ค.29 5985:48 /var/www/java/jre1.7.0/bin/java -Djava.util.logging.config.file=/v
ar/www/tokenize-7020/conf/logging.properties -Xms1024m -Xmx10246m -XX:NewSize=256m -XX:MaxNewSize=356m -XX:PermSize=256m -XX:MaxPermS
ize=1024m -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djava.endorsed.dirs=/var/www/tokenize-7020/endorsed -cla
sspath /var/www/tokenize-7020/bin/bootstrap.jar:/var/www/tokenize-7020/bin/tomcat-juli.jar -Dcatalina.base=/var/www/tokenize-7020 -Dc
atalina.home=/var/www/tokenize-7020 -Djava.io.tmpdir=/var/www/tokenize-7020/temp org.apache.catalina.startup.Bootstrap start
root     20030  0.0  0.0 113124  3152 ?        Ss   09:49   0:00 bash -c ps aux | grep "7020" && exit
root     20034  0.0  0.0 112656  2296 ?        S    09:49   0:00 grep 7020


172.17.101.238
tomcat    7023  0.0 15.5 14159240 1274128 ?     Sl   ส.ค.19 436:04 /var/www/java/jre1.7.0/bin/java -Djava.util.logging.config.file=/va
r/www/tokenize-7020/conf/logging.properties -Xms1024m -Xmx10246m -XX:NewSize=256m -XX:MaxNewSize=356m -XX:PermSize=256m -XX:MaxPermSi
ze=1024m -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djava.endorsed.dirs=/var/www/tokenize-7020/endorsed -clas
spath /var/www/tokenize-7020/bin/bootstrap.jar:/var/www/tokenize-7020/bin/tomcat-juli.jar -Dcatalina.base=/var/www/tokenize-7020 -Dca
talina.home=/var/www/tokenize-7020 -Djava.io.tmpdir=/var/www/tokenize-7020/temp org.apache.catalina.startup.Bootstrap start
root     14946  0.0  0.0 113124  3188 ?        Ss   ส.ค.23   0:00 bash -c ps aux | grep "7020" && exit
root     14950  0.0  0.0 112652  2364 ?        R    ส.ค.23   0:00 grep 7020
```

ภาพที่ 4.10 ตัวอย่างโพรเซสที่ทำงานอยู่บนหมายเลขไอพีแต่ละเครื่อง

จากภาพที่ 4.10 จะเห็นว่า หมายเลขไอพี 172.17.101.237 และ 172.17.101.238 กำลังรัน Tool ที่ใช้ในการทำ Sentence Segment อยู่



ภาพที่ 4.11 ตัวอย่างไฟล์สรุปผลของแต่ละหมายเลขไอพี

จากภาพที่ 4.11 จะเห็นว่า หมายเลขไอพี 209.126.119.65 มีค่า Process เป็น False ในขั้นตอนของการทำ Tokenize ซึ่งตัวโปรแกรมก็จะส่งอีเมลแจ้งเตือน ดังภาพที่ 4.12

ภาพที่ 4.12 ตัวอย่างอีเมลเมื่อมี Tool บนเครื่องเซิร์ฟเวอร์ใดไม่ทำงาน Sentence Segment หรือ Tokenize

4.5 การทำความสะอาดข้อมูลก่อนนำไปประมวลผล

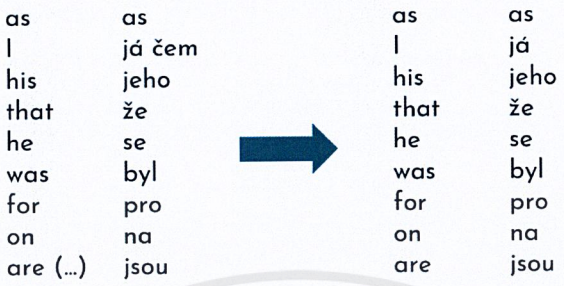
โปรแกรมนี้เป็นโปรแกรมสำหรับทำความสะอาดไฟล์ข้อมูล (ลบอักขระพิเศษที่ไม่ต้องการออก) ที่ได้มาจากอินเทอร์เน็ตหลายเว็บไซต์และจากเว็บไซต์ที่ลูกค้าต้องการ เพื่อนำมาทำ Dictionary ก่อนที่จะนำไปเป็นผลิตภัณฑ์ Dictionary สำหรับลูกค้าที่ต้องการ

ในโปรแกรมนี้ผู้จัดทำได้ทดสอบกับไฟล์ที่ทำการ c#Crawling มาจริง จำนวน 3 ไฟล์ ไฟล์ละ 1,000 บรรทัด และทดสอบโปรแกรมโดยแสดงเวลาเริ่มของแต่ละกระบวนการ คือ แยกคอลัมน์ของไฟล์ ลบอักขระพิเศษ และรวมคอลัมน์ของไฟล์ให้เป็นคู่ภาษาดังเดิม ได้ผลลัพธ์ ดังนี้

```
thapanee@DESKTOP-TR7DVSC:/mnt/d/work2.2561/project_coop/script/cleanDict/cleanDict$ bash split_clean_merge.sh dict/
Thu Dec 12 15:15:21 DST 2019
split process
Thu Dec 12 15:15:22 DST 2019
clean process
Thu Dec 12 15:15:55 DST 2019
merge process
Thu Dec 12 15:15:57 DST 2019
done
thapanee@DESKTOP-TR7DVSC:/mnt/d/work2.2561/project_coop/script/cleanDict/cleanDict$
```

ภาพที่ 4.13 ผลลัพธ์ที่ได้จากโปรแกรมทำความสะอาดข้อมูลก่อนนำไปประมวลผล

จากภาพที่ 4.13 จะเห็นว่า เวลาที่ใช้ขึ้นอยู่กับขั้นตอนของการลบอักขระพิเศษ ซึ่งเวลาจะแปรผันตรงกับขนาดของไฟล์ ซึ่งไฟล์ที่ใช้ทดสอบมีบรรทัดของไฟล์รวมกันที่ 3,000 บรรทัด ใช้เวลาในการทำงานทั้งสิ้น 36 วินาที โดยเฉลี่ยประมวลผลอยู่ที่ 83 บรรทัดต่อ 1 วินาที



ภาพที่ 4.14 ตัวอย่างผลลัพธ์ที่ได้หลังผ่านการทำความสะอาดข้อมูลแล้ว

จากภาพที่ 4.14 จะเห็นว่าเมื่อผ่านการทำความสะอาดข้อมูลแล้ว ในฝั่งคอลัมน์ที่ 1 มีตัวอักษร (...) ที่หายไป และฝั่งคอลัมน์ที่ 2 มี cem ที่หายไป เนื่องจากทั้งสองสิ่งที่มีระบุอยู่ในไฟล์รูปแบบที่ต้องการลบ

4.6 การตรวจสอบและหาชื่อไฟล์ที่ยังไม่ถูกประมวลผล

โปรแกรมนี้เป็นโปรแกรมสำหรับหาชื่อไฟล์ใน 2 โฟลเดอร์ระหว่างโฟลเดอร์เริ่มต้นและโฟลเดอร์หลังถูกประมวลผลแล้ว ว่ามีไฟล์ใดในโฟลเดอร์เริ่มต้นยังไม่ถูกประมวลผลหรือไม่

ในโปรแกรมนี้ผู้จัดทำได้ทดสอบกับไฟล์ในโฟลเดอร์เริ่มต้น จำนวน 10 ไฟล์ และไฟล์หลังถูกประมวลผลแล้ว ซึ่งชื่อไฟล์จะลงท้ายด้วย _HENC.srt ผลลัพธ์ที่ได้มีความถูกต้อง และสามารถตรวจหาไฟล์ที่ตกหล่นหรือไม่ถูกประมวลผลได้อย่างรวดเร็ว ช่วยให้แก้ไขได้อย่างทันท่วงทีและไม่มีไฟล์ใดที่สูญหายไปอย่างเปล่าประโยชน์ โดยรายชื่อไฟล์ที่ตกหล่นจากการประมวลผลจะบันทึกอยู่ในไฟล์ notpair.txt ดังภาพที่ 4.15, 4.16 และ 4.17

```
thapanee@DESKTOP-TR7DVSC:/mnt/d/work2.2561/project_coop/script/checkfileinHENC/checkfileinHENC/clean$ ll
total 0
drwxrwxrwx 1 thapanee thapanee 4096 Dec 11 17:38 ./
drwxrwxrwx 1 thapanee thapanee 4096 Dec 11 17:41 ../
-rwxrwxrwx 1 thapanee thapanee    0 Dec 11 17:38 Younger_S010_E09EN.srt*
-rwxrwxrwx 1 thapanee thapanee    0 Dec 11 17:38 Younger_S01_E09EN.srt*
-rwxrwxrwx 1 thapanee thapanee    0 Dec 11 17:38 Younger_S02_E09EN.srt*
-rwxrwxrwx 1 thapanee thapanee    0 Dec 11 17:38 Younger_S03_E09EN.srt*
-rwxrwxrwx 1 thapanee thapanee    0 Dec 11 17:38 Younger_S04_E09EN.srt*
-rwxrwxrwx 1 thapanee thapanee    0 Dec 11 17:38 Younger_S05_E09EN.srt*
-rwxrwxrwx 1 thapanee thapanee    0 Dec 11 17:38 Younger_S06_E09EN.srt*
-rwxrwxrwx 1 thapanee thapanee    0 Dec 11 17:38 Younger_S07_E09EN.srt*
-rwxrwxrwx 1 thapanee thapanee    0 Dec 11 17:38 Younger_S08_E09EN.srt*
-rwxrwxrwx 1 thapanee thapanee    0 Dec 11 17:38 Younger_S09_E09EN.srt*
```

ภาพที่ 4.15 ตัวอย่างไฟล์ในโฟลเดอร์เริ่มต้น

```
thapanee@DESKTOP-TR7DVSC:/mnt/d/work2.2561/project_coop/script/checkfileinHENC/checkfileinHENC/HENC$ ll
total 0
drwxrwxrwx 1 thapanee thapanee 4096 Dec 11 17:40 ./
drwxrwxrwx 1 thapanee thapanee 4096 Dec 11 17:41 ../
-rwxrwxrwx 1 thapanee thapanee    0 Dec 11 17:39 Younger_S01_E09EN_HENC.srt*
-rwxrwxrwx 1 thapanee thapanee    0 Dec 11 17:39 Younger_S02_E09EN_HENC.srt*
-rwxrwxrwx 1 thapanee thapanee    0 Dec 11 17:39 Younger_S03_E09EN_HENC.srt*
-rwxrwxrwx 1 thapanee thapanee    0 Dec 11 17:39 Younger_S04_E09EN_HENC.srt*
-rwxrwxrwx 1 thapanee thapanee    0 Dec 11 17:39 Younger_S05_E09EN_HENC.srt*
-rwxrwxrwx 1 thapanee thapanee    0 Dec 11 17:39 Younger_S06_E09EN_HENC.srt*
-rwxrwxrwx 1 thapanee thapanee    0 Dec 11 17:39 Younger_S07_E09EN_HENC.srt*
-rwxrwxrwx 1 thapanee thapanee    0 Dec 11 17:39 Younger_S08_E09EN_HENC.srt*
-rwxrwxrwx 1 thapanee thapanee    0 Dec 11 17:39 Younger_S09_E09EN_HENC.srt*
```

ภาพที่ 4.16 ตัวอย่างไฟล์ในโฟลเดอร์หลังถูกประมวลผล

```
thapanee@DESKTOP-TR7DVSC:/mnt/d/work2.2561/project_coop/script/checkfileinHENC/checkfileinHENC$ bash checkV2.sh clean/ HENC/
thapanee@DESKTOP-TR7DVSC:/mnt/d/work2.2561/project_coop/script/checkfileinHENC/checkfileinHENC$ cat notpair.txt
clean//Younger_S010_E09EN.srt
```

ภาพที่ 4.17 ผลลัพธ์ที่ได้จากโปรแกรมตรวจสอบและหาชื่อไฟล์ที่ยังไม่ถูกประมวลผล

4.7 การแยกข้อความในไฟล์โดยภาษา

โปรแกรมนี้เป็นโปรแกรมสำหรับแยกข้อความในไฟล์ 1 ไฟล์ที่มีการผสมกันระหว่าง 2 ภาษา เพื่อแยกเป็น 2 ไฟล์ คือ ไฟล์ที่เป็นภาษาอังกฤษ และ ไฟล์ที่เป็นคู่ภาษา เพื่อนำไปประมวลผลต่อไป

ในโปรแกรมนี้ผู้จัดทำได้ทดสอบกับไฟล์จากลูกค้าจริง ซึ่งเป็นคู่ภาษากันระหว่างภาษาอังกฤษและภาษาจีน โดยทำการทดสอบจำนวน 5 ไฟล์ แต่ละไฟล์มีชื่อ จำนวนบรรทัดและขนาดไฟล์ ดังนี้

ตารางที่ 4.3 แสดงความสัมพันธ์ระหว่างชื่อ จำนวนบรรทัดและขนาดของไฟล์ที่นำมาใช้ทดสอบโปรแกรมแยกข้อความในไฟล์โดยภาษา

ชื่อไฟล์	จำนวนบรรทัด	ขนาดไฟล์ (KB)
E_E_Administrative_Measures_for_the_Registration_of_Individually-owned_Businesses__Revised_in_2019_.txt	346	33
Law_of_the_People_s_Republic_of_China_on_Enterprise_Income_Tax__Revised_in_2018_.txt	485	42
Measures_of_ConsumersS_E_Association_on_Regulatory_Talks_with_Business_Operators_for_Protection_of_Rights_an__1_.txt	149	17
S_E_Administrative_Measures_for_Tax_Receipt__Revised_in_2019_.txt	474	70
S_E_Administrative_Measures_for_Tax_Registration__Revised_in_2019_.txt	360	43

เมื่อเปิดไฟล์ จะเห็นว่ามีทั้งบรรทัดว่าง บรรทัดที่เป็นภาษาอังกฤษ บรรทัดที่เป็นภาษาจีน และบางบรรทัดที่มีทั้งภาษาอังกฤษและภาษาจีนอยู่ร่วมกัน ดังภาพที่ 4.18

E_E_Administrative_Measures_for_the_Registration_of_Individually-owned_Businesses__Revised_in_2019_.txt

```
1  个体工商户登记管理办法(2019年修订)
2
3  Administrative Measures for the Registration of Individually-owned Businesses (Revised in 2019)
4
5
6
7  发文日期：2019-08-08
8
9  Promulgation date: 2019-08-08
10
11 地域：全国
12
13 Effective region: NATIONAL
14
15 颁布机关：国家市场监督管理总局
16
17 Promulgator: State Administration for Market Regulation
18
19 文号：国家市场监督管理总局令第14号
20
21 Document no:  Order of the State Administration for Market Regulation No.14
22
23 时效性：现行有效
24
25 Effectiveness: Effective
26
27 生效日期：2019-08-08
28
29 Effective date: 2019-08-08
30
31 所属产品分类：登记管理 （ 工商管理法->登记管理 ）
32
33 Category: Registration Administration ( Business Administration Law->Registration Administration )
```

ภาพที่ 4.18 ตัวอย่างเนื้อความในไฟล์อินพุต

```
thapanee@DESKTOP-TR7DVSC:/mnt/d/work 2.2561 (year3)/project coop/script/concat/concat$ bash separate_lang.sh input/
Wed Dec 11 11:46:07 DST 2019
input//E_E_Administrative_Measures_for_the_Registration_of_Individually-owned_Businesses__Revised_in_2019_.txt
Wed Dec 11 11:46:29 DST 2019
input//Law_of_the_People_s_Republic_of_China_on_Enterprise_Income_Tax__Revised_in_2018_.txt
Wed Dec 11 11:47:02 DST 2019
input//Measures_of_ConsumersS_E_Association_on_Regulatory_Talks_with_Business_Operators_for_Protection_of_Rights_an__1_.txt
Wed Dec 11 11:47:11 DST 2019
input//S_E_Administrative_Measures_for_Tax_Receipt__Revised_in_2019_.txt
Wed Dec 11 11:47:41 DST 2019
input//S_E_Administrative_Measures_for_Tax_Registration__Revised_in_2019_.txt
Wed Dec 11 11:48:03 DST 2019
thapanee@DESKTOP-TR7DVSC:/mnt/d/work 2.2561 (year3)/project coop/script/concat/concat$
```

ภาพที่ 4.19 เวลาเริ่มประมวลผลในแต่ละไฟล์

ผู้จัดทำได้ทดสอบโดยการประมวลผลไฟล์ทั้ง 5 ไฟล์ที่อยู่ในโฟลเดอร์ชื่อว่า Input โดยให้แสดงเวลาที่เริ่มประมวลผลแต่ละไฟล์ พร้อมแสดงชื่อไฟล์ ดังภาพที่ 4.19 ซึ่งสามารถสรุปเวลาในการประมวลผลทั้ง 5 ไฟล์ ดังตารางที่ 4.4

ตารางที่ 4.4 แสดงความสัมพันธ์ระหว่างชื่อไฟล์ จำนวนบรรทัด และเวลาที่ใช้ประมวลผล

ชื่อไฟล์	จำนวนบรรทัด	เวลา (วินาที)
E_E_Administrative_Measures_for_the_Registration_of _Individually-owned_Businesses__Revised_in_2019_.txt	346	22
Law_of_the_People_s_Republic_of_China_on_Enterprise _Income_Tax__Revised_in_2018_.txt	485	33
Measures_of_ConsumersS_E_Association_on_Regulatory _Talks_with_Business_Operators_for_Protection_of_Rights _an__1_.txt	149	9
S_E_Administrative_Measures_for_Tax_Receipt__Revised _in_2019_.txt	474	30
S_E_Administrative_Measures_for_Tax_Registration __Revised_in_2019_.txt	360	22

เมื่อประมวลผลทั้ง 5 ไฟล์สำเร็จแล้ว ผลลัพธ์ที่ได้จะอยู่ที่โฟลเดอร์ชื่อว่า Output โดยไฟล์อินพุต 1 ไฟล์ จะได้ไฟล์เอาต์พุต 2 ไฟล์ โดยใช้ชื่อไฟล์อินพุตตามด้วย _en.txt หรือ _zh.txt ซึ่งไฟล์เอาต์พุตจะเป็นไฟล์ที่แยกภาษาแล้ว ดังตัวอย่างภาพที่ 4.20 และ 4.21

E_E_Administrative_Measures_for_the_Registration_of_Individually-owned_Businesses__Revised_in_2019__en.txt

```
2  Promulgation date: 2019-08-08
3  Effective region: NATIONAL
4  Promulgator: State Administration for Market Regulation
5  Document no:  Order of the State Administration for Market Regulation No.14
6  Effectiveness: Effective
7  Effective date: 2019-08-08
8  Category: Registration Administration ( Business Administration Law->Registration Administration )
```

ภาพที่ 4.20 ตัวอย่างไฟล์เอาต์พุตฝั่งภาษาอังกฤษ

E_E_Administrative_Measures_for_the_Registration_of_Individually-owned_Businesses__Revised_in_2019__zh.txt

```
1  个体工商户登记管理办法(2019年修订)
2  发文日期：2019-08-08
3  地域：全国
4  颁布机关：国家市场监督管理总局
5  文号：国家市场监督管理总局令第14号
6  时效性：现行有效
7  生效日期：2019-08-08
8  所属产品分类：登记管理 ( 工商管理法->登记管理 )
```

ภาพที่ 4.21 ตัวอย่างไฟล์เอาต์พุตฝั่งภาษาจีน

4.8 การตรวจสอบและลบข้อความภาษาจีนแบบดั้งเดิม (Traditional) ออกจากไฟล์

โปรแกรมนี้เป็นโปรแกรมสำหรับตรวจสอบไฟล์ว่าเป็นภาษาจีนแบบดั้งเดิมหรือภาษาจีนแบบย่อ หากพบว่าเป็นภาษาจีนแบบดั้งเดิม จะทำการลบข้อความนั้นทิ้งทั้งบรรทัด (ทั้งฝั่งภาษาอังกฤษ และภาษาจีนที่เป็นคู่ภาษา)

ในโปรแกรมนี้ผู้จัดทำได้ทดสอบกับไฟล์ที่ตัดบางส่วนมาจากไฟล์ลูกค้าจริงจำนวน 1 ไฟล์ ที่มี 5,000 บรรทัด โดยให้โปรแกรมแยกไฟล์ (Split) ออกเป็นไฟล์ย่อยจำนวนไฟล์ละ 5,000 บรรทัด และทำงานพร้อมกันหลายไฟล์ตามแต่จะกำหนดในส่วนของ Max_jobs โดยผู้จัดทำเลือก Max_jobs อยู่ที่ 5 ไฟล์ ได้ผลการทดลอง ดังนี้

```
thapanee@DESKTOP-TR7DVSC:/mnt/d/work2.2561/project_coop/script/ENZH$ bash detectZH.sh 0.input/AstroData.EN-ZH_ZH.txt
Fri Dec 13 17:21:34 DST 2019
done
Fri Dec 13 17:32:59 DST 2019
```

ภาพที่ 4.22 เวลาที่ใช้ในการตรวจสอบและลบข้อความภาษาจีนแบบดั้งเดิมออกจากไฟล์

จากภาพที่ 4.22 จะเห็นว่า ระยะเวลาของโปรแกรมในการประมวลผลไฟล์ 5,000 บรรทัด ใช้เวลา 11:25 นาที หรือ 685 วินาที คิดเป็น 7 บรรทัดต่อ 1 วินาที

ZH.sh | AstroData.EN-ZH_ZH.txt.split0002.txt | AstroData.EN-ZH_EN.txt | AstroData.EN-ZH_ZH.txt

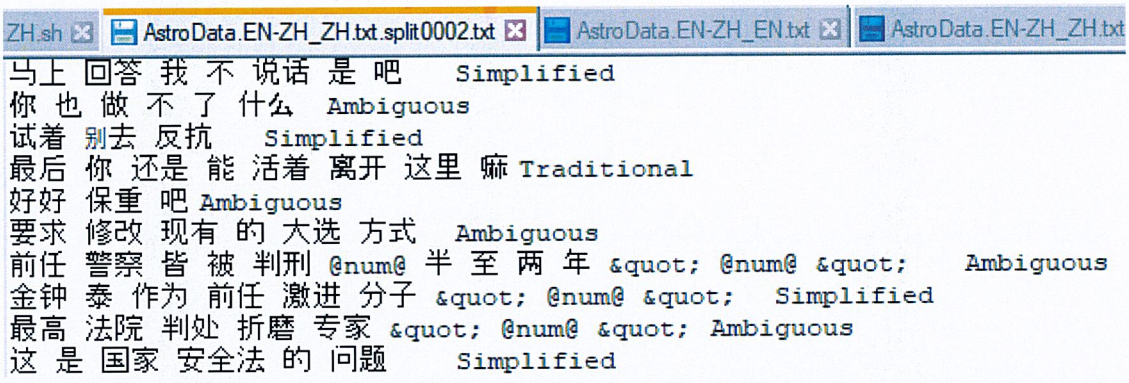
```
Answer me right away !
There &apos;s nothing you can do .
Just don &apos;t resist too much .
Ultimately , you get out of here alive .
Take good care of yourself .
Demanding the amendment of an election system .
Former policemen were sentenced separately to @num@ years and @num@ year and half ...
Kim Jong @-@ tae with activist background is elected to the @num@ Congress ...
The Supreme Court sentenced the torture expert Lee Doo @-@ han to spend @num@ years in prison .
It is a National Security Law issue .
```

ภาพที่ 4.23 ตัวอย่างไฟล์อินพุตฝั่งภาษาอังกฤษ

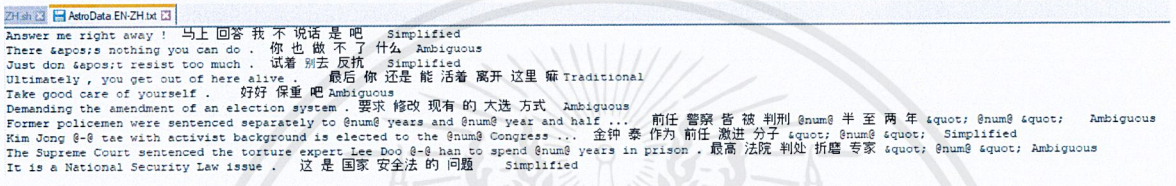
ZH.sh | AstroData.EN-ZH_ZH.txt

```
马上 回答 我 不 说话 是 吧
你 也 做 不 了 什么
试着 别去 反抗
最后 你 还是 能 活着 离开 这里 嘛
好好 保重 吧
要求 修改 现有 的 大选 方式
前任 警察 皆 被 判刑 @num@ 半 至 两 年 &quot; @num@ &quot;
金钟 泰 作为 前任 激进 分子 &quot; @num@ &quot;
最高 法院 判处 折磨 专家 &quot; @num@ &quot;
这 是 国家 安全法 的 问题
```

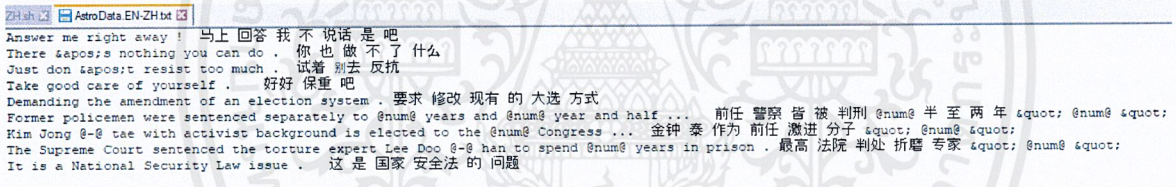
ภาพที่ 4.24 ตัวอย่างไฟล์อินพุตฝั่งภาษาจีน



ภาพที่ 4.25 ตัวอย่างไฟล์เมื่อผ่านการตรวจสอบว่าเป็นภาษาจีนแบบใด



ภาพที่ 4.26 ตัวอย่างไฟล์เมื่อผ่านการรวมกันของฝั่งภาษาอังกฤษและจีน



ภาพที่ 4.27 ผลลัพธ์สุดท้ายที่ได้จากโปรแกรม

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

โปรแกรมทั้ง 8 โปรแกรมมีวัตถุประสงค์ที่แตกต่างกัน ขึ้นอยู่กับทีมว่ามีปัญหาใด และต้องการได้ผลลัพธ์อย่างไร ซึ่งจากการทดสอบโปรแกรมทั้ง 8 โปรแกรม สามารถสรุปผลการทดลองและข้อเสนอแนะได้ 8 ข้อตามจำนวนโปรแกรม ดังนี้

1. โปรแกรมนับคำ ประโยค บรรทัด ย่อหน้าและรูปภาพในไฟล์

โปรแกรมนี้ถูกสร้างขึ้นเพื่อเป็นส่วนหนึ่งในผลิตภัณฑ์สำหรับขายลูกค้าในการนับคำ ประโยค บรรทัด ย่อหน้าและรูปภาพในไฟล์ 5 ประเภท คือ Document (.doc และ .docx), Power point (.ppt และ .pptx), Excel (.xls และ .xlsx), Text และ Portable Document Format (PDF)

จากการทดสอบ ผลลัพธ์ของโปรแกรมอยู่ในระดับความถูกต้องที่ 99% เนื่องจากบางส่วนของโปรแกรมคอมพิวเตอร์ใช้วิธีการ Regular Expression ซึ่งไม่สามารถครอบคลุมความถูกต้องได้ 100% โดยเวลาที่ใช้ในการประมวลผลแปรผันตรงตามขนาดของไฟล์

ข้อเสนอแนะสำหรับโปรแกรมนี้ คือ สามารถนำโปรแกรมไปต่อยอดในการนับภาษาอื่นที่นอกเหนือจากภาษาอังกฤษได้อีก และสามารถพัฒนาโปรแกรมให้มีความถูกต้อง 100% ได้

2. โปรแกรมตรวจสอบจำนวนคู่ภาษา

โปรแกรมนี้ถูกสร้างขึ้นเพื่อตรวจสอบไฟล์ XML ว่ามีจำนวนของคู่ภาษาตรงกันหรือไม่

จากการทดสอบ โปรแกรมสามารถทำงานได้อย่างถูกต้อง 100% และใช้เวลาในการประมวลผลเพียงเล็กน้อย โดยเฉลี่ยประมวลผลอยู่ที่ 967,811 บรรทัดของไฟล์ต่อ 1 วินาที

3. โปรแกรมตรวจสอบอายุของไฟล์และวันที่มีการปรับปรุงไฟล์ล่าสุด

โปรแกรมนี้ถูกสร้างขึ้นเพื่อหาไฟล์นามสกุลที่ต้องการกรณีที่ไม่รู้เฉพาะเจาะจงว่าอยู่ที่ Path ใด พร้อมบอกอายุและวันที่มีการปรับปรุงไฟล์ล่าสุด

จากการทดสอบ โปรแกรมสามารถทำงานได้อย่างถูกต้อง 100% และใช้เวลาในการประมวลผลเพียงเล็กน้อย

4. โปรแกรมตรวจสอบการทำงานของเครื่องเซิร์ฟเวอร์ที่ใช้ในการทำงานในระบบ

โปรแกรมนี้ถูกสร้างขึ้นเพื่อคอยตรวจสอบเครื่องเซิร์ฟเวอร์ว่า Tool สำหรับ Sentence Segment/Tokenize ทำงานอยู่หรือไม่ หาก Tool ทำงานอยู่ต้องตรวจสอบว่า Service ทำงานอยู่หรือไม่ หากพบข้อผิดพลาดจะส่งอีเมลเพื่อแจ้งเตือน

จากการทดสอบ โปรแกรมสามารถทำงานได้อย่างถูกต้อง 100% และใช้เวลาในการประมวลผลเพียงเล็กน้อย

ข้อเสนอแนะสำหรับโปรแกรมนี้ คือ สามารถพัฒนาโปรแกรมให้สะดวกมากขึ้น เช่น สร้างเว็บแอปพลิเคชันเพื่อนำผลลัพธ์จากโปรแกรมมาบอกข้อผิดพลาดและแก้ไขข้อผิดพลาดโดยการ Restart เครื่องเซิร์ฟเวอร์อัตโนมัติเมื่อกดปุ่ม Restart บนหน้าเว็บ เป็นต้น

5. โปรแกรมทำความสะอาดข้อมูลก่อนนำไปประมวลผล

โปรแกรมนี้ถูกสร้างขึ้นเพื่อลบอักขระพิเศษออกจาก Dictionary เพื่อให้ได้คำศัพท์ที่ถูกต้อง เพื่อเป็นผลิตภัณฑ์สำหรับลูกค้า และเพื่อนำไปเพิ่มใน Dictionary สำหรับ Drop Folder

จากการทดสอบ โปรแกรมสามารถทำงานได้อย่างถูกต้อง 100% และใช้เวลาในการประมวลผลเพียงเล็กน้อย โดยเฉลี่ยประมวลผลอยู่ที่ 83 บรรทัดต่อ 1 วินาที

6. โปรแกรมตรวจสอบและหาชื่อไฟล์ที่ยังไม่ถูกประมวลผล

โปรแกรมนี้ถูกสร้างขึ้นเพื่อตรวจหาไฟล์ที่ตกหล่นจากการประมวลผล เพื่อไม่ให้ไฟล์ใดไฟล์หนึ่งสูญหายไปอย่างเปล่าประโยชน์

จากการทดสอบ โปรแกรมสามารถทำงานได้อย่างถูกต้อง 100% และใช้เวลาในการประมวลผลเพียงเล็กน้อย

7. โปรแกรมแยกข้อความในไฟล์โดยภาษา

โปรแกรมนี้ถูกสร้างขึ้นเพื่อแยกข้อความในไฟล์ที่มีภาษาฝั่งภาษาอังกฤษและคู่ภาษา (ในที่นี่ทำภาษาจีน) ออกเป็น 2 ไฟล์ ตามภาษา เพื่อนำไฟล์ที่ได้ไปประมวลผลต่อไป

จากการทดสอบ โปรแกรมสามารถทำงานได้อย่างถูกต้อง 100% โดยเฉลี่ยประมวลผลอยู่ที่ 83 บรรทัดต่อ 1 วินาที

8. โปรแกรมตรวจสอบและลบข้อความภาษาจีนแบบดั้งเดิมออกจากไฟล์

โปรแกรมนี้ถูกสร้างขึ้นเพื่อแยกข้อความในไฟล์ออกตามภาษาของไฟล์นั้นๆ และเพื่อทดสอบการทำงานของไฟล์ Detector.jar ของนักพัฒนาในทีมว่ามีความถูกต้องมากเพียงใด และใช้เวลาในการทำงานนานหรือไม่

จากการทดสอบ โปรแกรมสามารถทำงานได้อย่างถูกต้อง แต่ใช้เวลาค่อนข้างนาน ซึ่งความเร็วของการทำงานแปรผันตรงตามขนาดของไฟล์ โดยเฉลี่ยสามารถประมวลผลได้ 7 บรรทัด ต่อ 1 วินาที

จากโปรแกรมที่ได้พัฒนาขึ้นทั้ง 8 โปรแกรม เมื่อได้นำทุกโปรแกรมไปใช้งานจริง พบว่า แต่ละโปรแกรมมีส่วนช่วยให้การทำงานมีความสะดวกสบายมากยิ่งขึ้น และเมื่อมีโปรแกรมต่างๆเข้ามาช่วยในการทำงาน ทำให้สามารถลดระยะเวลาในการทำงานของทีมลงไปได้ เพราะเปลี่ยนจากการทำงานด้วยมนุษย์เป็นการทำงานด้วยเครื่อง นอกจากนี้ยังช่วยลดความผิดพลาดจากการทำงานของมนุษย์ (Human Error) ลงไปได้

เอกสารอ้างอิง

Suphakit Annoppornchai. (2560). Unix คืออะไร. [ออนไลน์]. เข้าถึงได้จาก : https://saixiii.com/what-is-unix/ (วันที่ค้นข้อมูล : 2 ธันวาคม 2562).

โครงสร้างของ Linux. (2556). [ออนไลน์]. เข้าถึงได้จาก : https://panjaphon.wordpress.com/2013/03/06/โครงสร้างของ-linux/ (วันที่ค้นข้อมูล : 2 ธันวาคม 2562).

Purinat P. (2561). Linux คืออะไร. [ออนไลน์]. เข้าถึงได้จาก : https://medium.com/@sprizebnz/linux-คืออะไร-33c21a854b6a? (วันที่ค้นข้อมูล : 2 ธันวาคม 2562).

Suphakit Annoppornchai. (2560). kernel คืออะไร. [ออนไลน์]. เข้าถึงได้จาก : https://saixiii.com/what-is-kernel/ (วันที่ค้นข้อมูล : 2 ธันวาคม 2562).

Suphakit Annoppornchai. (2560). Shell คืออะไร. [ออนไลน์]. เข้าถึงได้จาก : https://saixiii.com/what-is-shell-unix-linux/ (วันที่ค้นข้อมูล : 2 ธันวาคม 2562).

Aoo Studio. (2562). shell script (เชลล์ สคริปต์) คืออะไร. [ออนไลน์]. เข้าถึงได้จาก : https://aoostudio.com/single-blog.php?id=37&shell%20script%20(เชลล์%C2%A0สคริปต์)%20คืออะไร (วันที่ค้นข้อมูล : 2 ธันวาคม 2562).

Suphakit Annoppornchai. (2560). Shell Script คืออะไร. [ออนไลน์]. เข้าถึงได้จาก : https://saixiii.com/what-is-shell-script (วันที่ค้นข้อมูล : 2 ธันวาคม 2562).

JavaScript คืออะไร. (2560). [ออนไลน์]. เข้าถึงได้จาก : https://www.mindphp.com/คู่มือ/73-คืออะไร/2187-java-javascript-คืออะไร.html (วันที่ค้นข้อมูล : 2 ธันวาคม 2562).

Thai Programmer Association. (2559). NodeJS ตอนที่ 1 NodeJs คืออะไร. [ออนไลน์]. เข้าถึงได้จาก : https://www.thaiprogrammer.org/2016/02/nodejs-ตอนที่-1-nodejs-คืออะไร/ (วันที่ค้นข้อมูล : 2 ธันวาคม 2562

เอกสารอ้างอิง (ต่อ)

เจาะลึกกับ node.js แบบเริ่มต้นทำความรู้จัก. (2556). [ออนไลน์]. เข้าถึงได้จาก : http://meewebfree.com/site/nodejs/441-learn-about-node-js-with-basic-of-node-js. (วันที่ค้นข้อมูล : 2 ธันวาคม 2562).

ภาคผนวก ก

การเขียนโปรแกรมคอมพิวเตอร์สำหรับโปรแกรมตรวจสอบจำนวนคู่ภาษา

```
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.stream.XMLInputFactory;
import javax.xml.stream.XMLStreamException;
import javax.xml.stream.XMLStreamReader;
import javax.xml.transform.TransformerConfigurationException;
import javax.xml.transform.TransformerException;
import org.xml.sax.SAXException;

public class CountLangTUV {
    public static void main(String[] args) throws FileNotFoundException, XMLStreamException, IOException,
ParserConfigurationException, SAXException, TransformerConfigurationException, TransformerException {
        int summary = 0;
        String input = "C:\\Users\\59010362\\Desktop\\input";
        File dir = new File(input);
        for (File file : dir.listFiles()) {
            try {
                XMLInputFactory factory = XMLInputFactory.newInstance();
                XMLStreamReader streamReader = factory.createXMLStreamReader(new FileReader(file));

                int count = 0;
                // Create a HashMap
                HashMap<String, Integer> map = new HashMap<>();

                while (streamReader.hasNext()) {
                    //Move to next event
                    streamReader.next();
                    //Check if its 'START_ELEMENT'
                    if (streamReader.getEventType() == XMLStreamReader.START_ELEMENT) {
                        if (streamReader.getLocalName().equalsIgnoreCase("tuv")) {
                            String lang = streamReader.getAttributeValue(null, "lang");
                            if (!map.containsKey(lang)) {
                                map.put(lang, 1);
```

```
                    } else {
                        // Get the iterator over the HashMap
                        Iterator<Map.Entry<String, Integer>> iterator = map.entrySet().iterator();

                        // Iterate over the HashMap
                        while (iterator.hasNext()) {

                            // Get the entry at this iteration
                            Map.Entry<String, Integer> entry = iterator.next();

                            // Check if this key is the required key
                            if (lang.equals(entry.getKey())) {
                                map.put(lang, map.get(lang) + 1);
                            }
                        }
                    }
                }
            }
        }
        if (map.values().toArray()[0] == map.values().toArray()[1]) {
            count = 2*(int) map.values().toArray()[0];
        }
        else {
            int a = (int) map.values().toArray()[0];
            int b = (int) map.values().toArray()[1];
            count = (a < b) ? 2*a : 2*b ;
        }
        System.out.println(file.getName() + "\t\t" + count);
        System.out.println("   language: " + map + "\n");
        summary += count;

    } catch (FileNotFoundException | XMLStreamException e) {
    }
  }
  System.out.println("summary " + summary);
 }
}
```

ภาคผนวก ข

การเขียนโปรแกรมคอมพิวเตอร์สำหรับโปรแกรมตรวจสอบอายุของไฟล์และวันที่มีการปรับปรุงไฟล์ล่าสุด

```
rm -f last_modified.txt error_last_modified.txt age.txt error_age.txt
input=$1
while read line
do
        pathinline=$(echo $line | cut -d"=" -f 2 | sed -r 's/[]+/ /g')
        path=$(echo $pathinline | cut -d" " -f 1)
        type=$(echo $pathinline | cut -d" " -f 2)
        list=$(find $path -type f -name "*.$type")
        for eachpath in $list
        do
                ftime=`stat -c %Y $eachpath` 2> error_age.txt
                ctime=`date +%s` 2> error_age.txt
                diff=$(( (ctime - ftime) / 86400 )) 2> error_age.txt
                echo $diff  $eachpath >> age.txt 2> error_age.txt
        done
        find $path -name "*.$type" -printf "%TD \t%p\n" >> last_modified.txt 2> error_last_modified.txt
done < $input
```

ภาคผนวก ค

การเขียนโปรแกรมคอมพิวเตอร์สำหรับโปรแกรมทำความสะอาดข้อมูลก่อนนำไปประมวลผล

1. โปรแกรมในส่วนของการแยกคอลัมน์ของไฟล์

```
input=$1
date; echo "split process"
mkdir -p split
for file in $(ls $input/*.txt);
do
        bn=$(basename $file)
        count_=$(echo $bn | awk -F'_' '{print NF-1}')
        if [[ $count_ -eq 1 ]];
        then
                lang=$(echo $bn | sed "s/.txt//g" | awk -F '_' '{print $2}')
                cat $file | awk -F '\t' '{print $1}' > ./split/EN_"$lang"_EN.txt
                cat $file | awk -F '\t' '{print $2}' > ./split/EN_"$lang"_"$lang".txt
        elif [[ $count_ -eq 2 ]]; then mv $file ./split/$bn; else echo "filename is wrong"; fi
done
```

2. โปรแกรมในส่วนของการลบอักขระพิเศษ

```
input=$1
date; echo "clean process"
pattern_for_clean="./pattern_for_clean.cfg"
PROCESS="./process"
FINISH="./finish"
Cleantxt="$PROCESS/clean.txt"
Outtxt="./$PROCESS/out.txt"
mkdir -p process finish
for file in $(ls $input/*.txt);
do
        name=$(basename "$file" | sed "s/.txt//g")
        target=$(echo $name | awk -F '_' '{print $3}')
        mv $file $PROCESS/clean.txt
        while read line
        do
                startwithAtoZ=$([[ `echo $line | grep "^[A-Z]"` ]] && echo yes || echo no )
                if [[ $startwithAtoZ == "yes" ]]
                then
                        lang=$(echo $line | grep ^[A-Z] | awk -F ' ' '{print $1}')
                        if [[ $lang == $target ]]
                        then
                                pattern=$(echo $line | awk -F ' ' '{print $2}'
```

```
                                        perl -pe  "s/$pattern/ /g" $Cleantxt > $Outtxt
                                fi
                        elif [[ $startwithAtoZ == "no" ]]; then perl -pe  "s/$line/ /g" $Cleantxt > $Outtxt ; fi
                        if [[ -f $Cleantxt && -f $Outtxt ]]; then rm -f $Cleantxt; mv $Outtxt $Cleantxt; fi
        done < $pattern_for_clean
        mv $Cleantxt $FINISH/"$name".txt
done
rm -rf process
```

3. โปรแกรมในส่วนของการรวมคอลัมน์ของไฟล์เป็นคู่ภาษาดังเดิม

```
input=$1
date; echo "merge process"
path="./finish/"
for file in $(ls $input/*EN.txt)
do
        name=$(basename "$file" | sed "s/.txt//g")
    target=$(echo $name | awk -F '_' '{print $2}')
        check=$(echo $name | awk -F '_' '{print $3}')
        for file2 in $(ls $input/*.txt)
        do
                name2=$(basename "$file2" | sed "s/.txt//g")
        target2=$(echo $name2 | awk -F '_' '{print $2}')
                check2=$(echo $name2 | awk -F '_' '{print $3}')
                if [[ ($target == $target2) && ($check != $check2) ]]
                then
                        paste "$path$name".txt "$path$name2".txt > ./output/EN_"$target".txt
                        break
                fi
        done
done
if [[ -f ./output/EN_EN.txt ]]; then rm -f ./output/EN_EN.txt; fi
```

ภาคผนวก ง

การเขียนโปรแกรมคอมพิวเตอร์สำหรับโปรแกรมตรวจสอบและหาชื่อไฟล์ที่ยังไม่ถูกประมวลผล

```
INPUTFOLDER=$1
OUTPUTFOLDER=$2
rm -f notpair.txt
for file in $(ls $1/*)
do
        fname=$(basename $file)
        name=$(echo $fname | awk -F'.' '{ print $1 }')
        check=$([[ `ls $2/* | grep "$name"` ]] && echo "true" || echo "false")
        if [[ $check == “false” ]]; then echo $file >> notpair.txt; fi
done
```

ภาคผนวก จ

การเขียนโปรแกรมคอมพิวเตอร์สำหรับโปรแกรมแยกข้อความในไฟล์โดยภาษา

```
folder=$1
for file in $(ls $folder/*.txt)
do
        echo $file
        sed '/^$/d' $file > newfile.txt
        name=$(basename $file | sed "s/.txt//g")
        while read line
        do
                check=$([[ `echo $line | grep -P "^[0-9]|^[A-Z]|^[a-z]|^[(\"\[]" | grep -P "[A-Z]$|[a-z]$|[.,;:\"]$|[0-9]$|[)\]]$"`
]] && echo "true" || echo "false")
                if [[ $check = "true" ]]
                then
                        echo "$line" >> ./output/"$name"_en.txt
                else
                        k=$(echo $line | grep -i "[a-z]")
                        if [[ $k ]]
                        then
                                zh=$(sed "s/[a-z].*$//gI" <<< $line)
                                if [[ $zh ]]
                                then
                                        echo $zh >> ./output/"$name"_zh.txt
                                        en=$(sed "s/$zh//gI" <<< $line)
                                        echo $en >> ./output/"$name"_en.txt
                                else
                                        echo $line >> ./output/"$name"_en.txt
                                fi
                        else
                                echo "$line" >> ./output/"$name"_zh.txt
                        fi
                fi
        done < newfile.txt
done
rm -f newfile.txt
```

ภาคผนวก ฉ

การเขียนโปรแกรมคอมพิวเตอร์สำหรับโปรแกรมตรวจสอบและลบข้อความภาษาจีนแบบดั้งเดิม (Traditional) ออกจากไฟล์

```
#! /usr/bin/env bash
date
set -o monitor   # means: run background processes in a separate processes...

function do_job {
        name=$(basename ${todo_array[$index]})
        while read line
        do
                ch=$(java -jar /mnt/d/work2.2561/project_coop/script/ENZH/detector.jar $line)
                check=$(echo $ch | awk -F' ' '{ print $4 }')
                printf "%s\t%s\n" "$line" "$check" >> /mnt/d/work2.2561/project_coop/script/ENZH/2.detect/$name
                done < ${todo_array[$index]}
        sleep 2
}

function add_next_job {
        if [[ $index -lt ${#todo_array[*]} ]]
        then
                do_job ${todo_array[$index]} &
                index=$(($index+1))
        fi
}

trap add_next_job CHLD  # execute add_next_job when we receive a child complete signal
inputTarget=$1
target=$(basename $1)
split --lines=10000 --numeric-suffixes=1 --suffix-length=4 --additional-suffix=.txt $1 ./1.split/$target.split
todo_array=($(ls ./1.split/*.txt))
index=0
max_jobs=5
while [[ $index -lt $max_jobs ]]; do add_next_job; done
wait  # wait for all jobs to complete
find ./2.detect -type f -name '*.txt' -exec cat {} + >> ./3.append/$target
EN=$(sed 's/ZH.txt/EN.txt/g' <<< $target)
ENZH=$(sed 's/_EN.txt/.txt/g' <<< $EN)
paste ./0.input/$EN ./3.append/$target > ./4.merge/$ENZH
sed '/Traditional/d' ./4.merge/$ENZH > ./5.output/$ENZH
echo "done"; date
```