



1.1 ความเป็นมาของปัญหา

1.1.1 ระบบการจัดรูปแบบของข้อความในเอกสาร (Text Processing)

ระบบการจัดรูปแบบของข้อความในเอกสารมีขั้นตอนการทำงาน 3 ขั้นตอน คือ การนำข้อมูลเข้าและตรวจสอบ การจัดรูปแบบข้อความ และการแสดงผลลัพธ์

ระบบการจัดรูปแบบข้อความในเอกสาร แบ่งออกได้เป็น 2 ลักษณะตามการทำงาน คือ การจัดรูปแบบคำ (word formatting) และการประมวลผลคำ (word processing)

การจัดรูปแบบคำเริ่มใช้ครั้งแรกในปี ค.ศ. 1964¹ สำหรับจัดรูปแบบของข้อความหรือเอกสารซึ่งมีภาษาสั่งงานอย่างง่าย เพื่อประมวลผลข้อความเหล่านั้นให้ได้รูปแบบตามที่ต้องการในการจัดแบบรูปแบบของข้อมูลนำเข้า (input) และผลที่ได้ (output) อาจมีลักษณะที่แตกต่างกันมาก เนื่องจากวิธีการทำงานจะไม่ทำการจัดรูปแบบของข้อความในทันทีที่นำข้อมูลเข้าแต่ต้องกำหนดรูปแบบของเอกสารที่ต้องการ แล้วเรียกใช้คำสั่งของการจัดรูปแบบรวมกันในเนื้อหาของข้อความที่ต้องการพิมพ์ ดังนั้นในข้อความที่นำเข้าจะมีทั้งคำสั่งของการจัดรูปแบบและเนื้อหาของข้อความรวมกัน เมื่อกำหนดคำสั่งและพิมพ์ข้อความที่ต้องการหมดแล้ว ต้องเรียกใช้โปรแกรมสำหรับจัดรูปแบบเอกสารอีกครั้งหนึ่ง จึงจะได้เอกสารตามรูปแบบที่ต้องการ ในปัจจุบันมีระบบการจัดรูปแบบคำที่ใช้กันแพร่หลายอยู่ 2 ระบบ คือ nroff/troff และ T_EX

การประมวลผลคำ เป็นการพัฒนากการจัดรูปแบบคำเพื่อให้มีการจัดรูปแบบของคำหรือข้อความในขณะที่ทำการป้อนข้อมูล ดังนั้นผลที่ปรากฏบนจอภาพคือผลสุดท้ายที่ผู้ใช้จะได้รับด้วยระบบการประมวลผลคำที่ใช้กันทั่วไปเช่น ed, ex, vi ซึ่งเป็นโปรแกรมบรรณาธิการที่ใช้ภายใต้ยูนิกซ์ นอกจากนี้ยังมีโปรแกรมบรรณาธิการสำหรับระบบปฏิบัติการอื่น ๆ อีกมากมาย ในที่นี้จะกล่าวถึงเฉพาะ vi ซึ่งเป็นโปรแกรมบรรณาธิการภายใต้ระบบปฏิบัติการยูนิกซ์เท่านั้น

1. Funita R., Scofield J. & Shaw A., Document Formatting Systems : Survey, Concepts, and Issues, ACM Computing Surveys, 1982, pp.422

ตัวอย่างของโปรแกรมการจัดรูปแบบข้อความในเอกสารที่ใช้ภายใต้ระบบปฏิบัติการยูนิกซ์ที่รู้จักกันโดยทั่วไป คือ vi และ nroff/troff

1.1.1.1 vi เป็นโปรแกรมบรรณาธิการที่ใช้ภายใต้ยูนิกซ์ มีลักษณะเป็นโปรแกรมบรรณาธิการบนจอภาพ (screen editor) สามารถแสดงข้อความบางส่วนของแฟ้มข้อมูลได้

ลักษณะงานที่ทำภายใต้ vi

1. การกำหนดขนาดของส่วนที่จะใช้แสดงข้อความบนจอภาพ สามารถกำหนดเป็นจำนวนบรรทัดต่อหนึ่งจอภาพ (set window)

2. การค้นหาข้อความที่ต้องการ (searching string)

3. การลบข้อความ (deleting text)

4. การแทนที่ข้อความเก่าด้วยข้อความใหม่ (replacing text)

5. การเพิ่มข้อความ (inserting text)

6. การสร้างแฟ้มข้อมูลใหม่ (creating new file)

7. การเก็บข้อความในแฟ้มข้อมูลเป็นการถาวร (saving text)

8. การแสดงสถานะของแฟ้มข้อมูล (file status)

9. การคัดลอก เคลื่อนย้ายข้อความ (copy text)

10. การรวมข้อความเข้าด้วยกัน (join text)

1.1.1.2 nroff เป็นโปรแกรมสำหรับจัดรูปแบบของข้อความ เพื่อให้ได้ผลลัพธ์ที่ต้องการ การจัดรูปแบบข้อความของเอกสารโดยใช้ nroff พอดีจะจำแนกกระบวนการสำคัญออกได้ดังนี้

1. การกำหนดรูปแบบตัวพิมพ์ (font type) เป็นการกำหนดรูปแบบตัวพิมพ์ต่างๆ และวิธีการพิมพ์ตัวอักษรที่เป็นผลลัพธ์

2. การจัดบรรทัด (line control) เป็นการกำหนดรูปแบบการพิมพ์แต่ละบรรทัด

3. การจัดหน้า (page layout) สำหรับเป็นการกำหนดรูปแบบการพิมพ์ในแต่ละหน้าของเอกสาร

4. การจัดพิมพ์ข้อความที่ใช้สำหรับเป็นขอบบนและขอบล่างของเอกสารในแต่ละหน้า (header and footer control)

5. การจัดการเกี่ยวกับการใช้ตัวอักษรพิเศษบางตัว (directive control) เพื่อช่วยในการจัดรูปแบบของเอกสาร โดยจะใช้ตัวอักษรพิเศษบางตัวที่มีความหมายเฉพาะใน nroff ซึ่งเรียกว่าอักขระหลัก (escape character) และในเอกสารที่ได้จากการพิมพ์นั้น

ตัวอักษรเหล่านี้จะไม่ปรากฏออกมา เพียงแต่เป็นตัวช่วยในการกำหนดรูปแบบของเอกสารตามที่ต้องการ

6. การจบงาน (termination) เป็นการกำหนดการสิ้นสุดการทำงานของ profile ที่จุดต่าง ๆ นอกเหนือจากการสิ้นสุดข้อความปกติของแฟ้มข้อมูล

1.1.2 ระบบการจัดรูปแบบข้อความภาษาไทย (Thai Text Formatting System)

ระบบการจัดรูปแบบข้อความภาษาไทย มีความแตกต่างจากการจัดข้อความภาษาอังกฤษ เนื่องจากลักษณะที่แตกต่างกันของภาษา ในภาษาไทยลักษณะการเขียนมีหลายระดับ ในขณะที่ในภาษาอังกฤษใช้เพียงระดับเดียวเท่านั้น นอกจากนี้ในภาษาไทยไม่สามารถแยกคำได้อย่างชัดเจน ด้วยวิธีการเขียน ดังนั้น เมื่อต้องการใช้ข้อความภาษาไทยเป็นข้อมูลของโปรแกรมจัดรูปแบบ โดยทั่วไปนั้นจึงไม่สามารถใช้โปรแกรมที่มีอยู่ได้ เพราะโปรแกรมเหล่านั้นจะรับตัวอักษรทุกตัวเป็นระดับเดียวกันหมด เมื่อนำภาษาไทยไปใช้กับโปรแกรมนี้ สระและพยัญชนะจะอยู่ในบรรทัดเดียวกันด้วย

จากสาเหตุดังกล่าว จึงต้องมีการสร้างโปรแกรมสำหรับจัดข้อความภาษาไทยโดยเฉพาะ เพื่อแก้ปัญหาที่เกิดขึ้น

1.1.3 ระบบปฏิบัติการยูนิกซ์ (UNIX Operating System)

ระบบปฏิบัติการยูนิกซ์พัฒนาขึ้นโดย Ken Thompson และ Dennis Ritchie ในประมาณปี.ศ. 1970² ที่ Bell Laboratories ในรัฐ New Jersey โดยมีวัตถุประสงค์เพื่อให้เป็นระบบปฏิบัติการที่เล็กและใช้งานได้สะดวก มีอุปกรณ์ช่วยเพิ่มประสิทธิภาพในการเขียนและพัฒนาโปรแกรมมาก ปัจจุบันยูนิกซ์เป็นระบบปฏิบัติการที่ใช้กันแพร่หลายมากที่สุดระบบหนึ่ง สามารถใช้ได้ทั้งในคอมพิวเตอร์ขนาดเล็ก เช่น IBM PC จนถึงเครื่องคอมพิวเตอร์ขนาดใหญ่ เช่น CRAY II เป็นที่คาดกันว่ายูนิกซ์จะเป็นระบบปฏิบัติการที่สำคัญที่สุดระบบหนึ่งในเครื่องคอมพิวเตอร์ขนาดเล็กและขนาดกลาง และขณะนี้กล่าวได้ว่า ยูนิกซ์เป็นระบบปฏิบัติการที่จะมีบทบาทมากในวงการคอมพิวเตอร์ของไทย

2. Bourne S. R., The UNIX System, Addison-Wesley Publishing Company, 1983, pp.2

1.3 ขอบเขตของการวิจัย

ภายใต้ระบบปฏิบัติการยูนิกซ์ การจัดรูปแบบของข้อความในเอกสาร เน้นหนักในด้าน การจัดรูปแบบคำมากกว่าการประมวลผลคำ และในการทำวิจัยครั้งนี้จะเน้นเฉพาะการจัด รูปแบบคำเท่านั้น ซึ่งขอบข่ายของงานที่จะทำมีดังนี้

1. tvi (Thai Visual Editor) จัดรูปแบบของการบรรณาธิกรภายใต้ยูนิกซ์ คือ vi
2. tnroff (Thai nroff) เป็นโปรแกรมจัดรูปแบบที่รับข้อความภาษาไทยภายใต้ ยูนิกซ์
3. Device Driver เป็นส่วนที่กำหนดการแสดงผลทางเครื่องพิมพ์ในลักษณะของการ แสดงผลถาวร (hard copy)

1.4 ขั้นตอนในการทำวิจัย

1. ศึกษาเกี่ยวกับระบบปฏิบัติการยูนิกซ์และการใช้งานภายใต้ยูนิกซ์
2. ศึกษาภาษา C
3. ศึกษาเกี่ยวกับการประมวลผลข้อความ (text processing)
4. ออกแบบระบบโปรแกรมการประมวลผลคำ
5. พัฒนาโปรแกรมที่ได้ออกแบบขึ้น
6. ทดสอบระบบและโปรแกรม
7. จัดทำรายงานการวิจัย

1.5 ประโยชน์ที่ได้รับจากการวิจัย

1. ได้เรียนรู้เกี่ยวกับระบบปฏิบัติการยูนิกซ์
2. เพื่อใช้เป็นโปรแกรมการจัดเอกสารภาษาไทย ที่ใช้ในเครื่องคอมพิวเตอร์ภายใต้ ระบบปฏิบัติการยูนิกซ์
3. ช่วยให้การผลิตและค้นหาเอกสารภาษาไทยเป็นไปอย่างรวดเร็ว สะดวกและมี ประสิทธิภาพสูง
4. สามารถนำไปพัฒนาใช้บนเครื่อง ไมโครคอมพิวเตอร์ที่ใช้ยูนิกซ์ในอนาคต

1.1.4 ระบบการจัดการจัดรูปแบบข้อความภาษาไทยภายใต้ระบบปฏิบัติการยูนิกซ์

(Thai Text Formatting System under UNIX)

จากสาเหตุที่กล่าวมาแล้วเกี่ยวกับปัญหาการใช้ภาษาไทยในคอมพิวเตอร์ทั่ว ๆ ไป ทำให้ต้องมีการปรับปรุงแก้ไขหรือสร้างโปรแกรมสำเร็จรูปเพื่อใช้สำหรับงานประมวลผลที่เป็นภาษาไทย เพื่อให้รับข้อมูลภาษาไทยและแสดงผลในลักษณะที่ถูกต้องตามหลักการเขียนด้วย และเนื่องจากยูนิกซ์ได้เข้ามามีบทบาทในประเทศไทยเมื่อไม่นานมานี้ ดังนั้นการพัฒนาโปรแกรมที่สามารถรับข้อความภาษาไทยภายใต้ยูนิกซ์จึงเป็นปัญหาเร่งด่วนที่ได้รับความสนใจเพราะเมื่อใช้ภาษาไทยเป็นข้อมูลนำเข้าของโปรแกรมบรรณาธิการของยูนิกซ์ ไม่สามารถจัดรูปแบบการเขียนได้ และนอกจากนี้ภายใต้ยูนิกซ์เองมีโปรแกรมอรรถประโยชน์ (utility program) ที่มีประโยชน์ในการผลิตเอกสารโปรแกรมหนึ่ง คือ โปรแกรมจัดรูปแบบที่เรียกว่า nroff/troff แต่อย่างไรก็ตามโปรแกรมเหล่านี้ใช้สำหรับการจัดข้อความของเอกสารที่เป็นภาษาอังกฤษเท่านั้น ไม่สามารถใช้กับภาษาไทยได้ เนื่องจากปัญหาของภาษาตามที่ได้กล่าวมาแล้วข้างต้น

เนื่องจากการผลิตเอกสารส่วนใหญ่เป็นภาษาไทย และการที่สามารถนำคอมพิวเตอร์มาใช้ในการผลิตเอกสารจำนวนมาก จะช่วยให้สะดวกรวดเร็วและมีประสิทธิภาพมากขึ้น แต่ในขณะนี้ไม่มีโปรแกรมสำหรับใช้ในงานลักษณะดังกล่าว โดยเฉพาะอย่างยิ่งภายใต้ระบบปฏิบัติการยูนิกซ์

จากสาเหตุที่กล่าวมานี้ จึงได้จัดทำการศึกษาโดยการสร้างโปรแกรมสำหรับจัดข้อความของเอกสารภาษาไทยภายใต้ระบบปฏิบัติการยูนิกซ์ขึ้น โดยสร้างโปรแกรมบรรณาธิการและโปรแกรมจัดรูปแบบ เพื่อช่วยให้การผลิตเอกสารสมบูรณ์เหมือนกับการใช้โปรแกรมผลิตเอกสารภาษาอังกฤษของยูนิกซ์

1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อสร้างโปรแกรมบรรณาธิการที่สามารถรับและแสดงผลข้อความภาษาไทยใต้ระบบปฏิบัติการยูนิกซ์
2. เพื่อสร้างโปรแกรมจัดรูปแบบเอกสารภาษาไทย
3. เพื่อสร้างโปรแกรมแสดงผลลัพธ์
4. เพื่อสร้างโปรแกรมค้นหาเอกสาร

เหล่านั้นเก็บไว้ในแฟ้มข้อมูลเป็นการถาวรต่อไป อาจมองบัฟเฟอร์ให้เปรียบเสมือนกระดานดำสำหรับเขียน ซึ่งเมื่อเขียนครั้งหนึ่งแล้วสามารถลบออกเพื่อใช้ในครั้งต่อไปได้อีก vi ได้นำคุณสมบัติเด่นนี้มาใช้ในการทำงาน ซึ่งบัฟเฟอร์แบ่งออกตามลักษณะการใช้งานคือ

3.3.1.1 บัฟเฟอร์ใช้งาน (Work Buffer) เป็นบัฟเฟอร์ที่ใช้สำหรับเก็บข้อความทั้งหมดที่ใช้ในการบรรณาธิกร ในที่นี้จะเรียกสั้นๆ ว่า บัฟเฟอร์ โดยที่เมื่อเรียกใช้ vi จะทำการคัดลอกข้อความจากแฟ้มข้อมูลที่กำหนดเก็บไว้ในบัฟเฟอร์นี้ จากนั้นจะทำงานทุกอย่างภายในบัฟเฟอร์ การแก้ไขเปลี่ยนแปลงข้อความขณะที่ใช้โปรแกรม vi ไม่มีผลต่อข้อความที่เก็บในแฟ้มข้อมูลจนกว่าจะมีคำสั่งให้เก็บข้อความในบัฟเฟอร์เข้าสู่แฟ้มข้อมูล แต่ถ้าไม่มีการกำหนดให้เก็บข้อความดังกล่าว จะไม่ทำการเปลี่ยนแปลงข้อความในแฟ้มข้อมูลที่เกี่ยวข้อง ดังนั้นข้อความในแฟ้มข้อมูลยังคงเป็นข้อความเดิมก่อนการใช้ vi นั่นเอง

3.3.1.2 บัฟเฟอร์ทั่วไป (General Buffer) นอกจากบัฟเฟอร์ที่กล่าวถึงแล้ว vi ยังนำคุณสมบัติเด่นของบัฟเฟอร์มาใช้ประโยชน์อื่นอีก คือ ใช้สำหรับเก็บข้อความที่ถูกเปลี่ยนแปลงเพื่อที่จะเรียกข้อความเดิมกลับคืนมา หรือเพื่อนำไปใช้งานในครั้งต่อไป เช่นในการแก้ไขหรือเพิ่มข้อความแต่ละครั้งจะทำการเก็บไว้ในบัฟเฟอร์ด้วย ซึ่งเรียกว่าบัฟเฟอร์ทั่วไป แต่เนื่องจากเป็นบัฟเฟอร์ที่ใช้ร่วมกันทุกครั้งที่มีการเปลี่ยนแปลงข้อความ ดังนั้นข้อความเดิมที่เก็บในบัฟเฟอร์ทั่วไปจะถูกแทนที่ด้วยข้อความที่เกิดขึ้นใหม่เสมอ เมื่อเรียกข้อความจากบัฟเฟอร์นี้มาใช้จะเป็นข้อความครั้งสุดท้าย ถ้าต้องการเก็บข้อความเหล่านั้นไว้ใช้หลายครั้ง จะต้องกำหนดให้เก็บในบัฟเฟอร์เฉพาะ ซึ่งทำได้โดยการกำหนดชื่อให้แก่บัฟเฟอร์เหล่านั้น

3.3.1.3 บัฟเฟอร์กำหนดชื่อ (Named Buffer) สามารถกำหนดบัฟเฟอร์ให้แตกต่างกันโดยการกำหนดชื่อให้แก่แต่ละบัฟเฟอร์ เพื่อใช้สำหรับการเก็บข้อความไว้ตลอดไป หรือจนกว่าจะกำหนดให้ลบออก ในการใช้บัฟเฟอร์ชนิดนี้สามารถกำหนดได้ 26 บัฟเฟอร์ โดยการกำหนดชื่อบัฟเฟอร์ด้วยตัวอักษร a-z และเมื่อเรียกใช้จะเรียกด้วยชื่อที่กำหนด

3.3.2 แบบการทำงาน (Mode of Operation) ใน vi จะแบ่งรูปแบบของการทำงานออกเป็น 2 ลักษณะคือ การรับคำสั่งงาน (command mode) และการแก้ไขข้อความ (edit mode) โดยในที่นี้การรับคำสั่งจะรับตัวอักษรทุกตัวเป็นคำสั่ง (command) ของ vi และคำสั่งเหล่านี้จะไม่ปรากฏให้เห็นบนจอภาพ แต่เมื่ออยู่ในช่วงของการแก้ไขข้อความ จะรับตัวอักษรทุกตัวเป็นข้อความที่จะเก็บในบัฟเฟอร์ ตามปกติเมื่อ vi เริ่มทำงานจะรอรับคำสั่งเพื่อนำไปประมวลผลก่อนเสมอ และเมื่อเข้าสู่ช่วงของการแก้ไขข้อความด้วยคำสั่งใดคำสั่งหนึ่งที่เกี่ยวข้อง เช่นคำสั่ง i หรือ a จึงจะรับข้อความเก็บในบัฟเฟอร์ ถ้าต้องการเรียกใช้คำสั่งอื่นในขณะที่อยู่