บทที่ 6

การแบ่งคำภาษาไทยในโปรแกรมซียูไรท์เตอร์

บทนำ

ในการเขียนข้อความภาษาไทยนั้น แต่ละคำในประโยคอาจจะเขียนต่อเนื่องกันไปโดยไม่มีช่องว่างระหว่างคำเหมือนภาษาอังกฤษ รวมทั้งยังมีคำยกเว้นอีกมากมายที่จะต้องให้คอมพิวเตอร์ได้จดจำ ซึ่งในลักษณะเช่นนี้ทำให้การประมวลผลคำภาษาไทยจึงประสบปัญหายุ่งยาก ในเรื่องของการแบ่งคำภาษาไทยเพื่อใช้ในการจัดช่วงหลังให้ด้านขวาของบรรทัดตรงกัน

การแบ่งคำภาษาไทย จะเกิดขึ้นเมื่อได้เขียนข้อความมาถึงตำแหน่งที่ต้องการให้แบ่งคำ ซึ่งอาจจะเป็นตำแหน่งด้านขอบขวาสุดของจอภาพก็ได้ โดยข้อความที่ถูกแบ่งหลังตำแหน่งที่ต้องการ จะไปขึ้นบรรทัดใหม่ให้เอง ทำให้สะดวกต่อการใช้งานมาก

6.1 ลักษณะของตัวอักษรภาษาไทย

จากการวิเคราะห์สถิติการใช้ตัวอักษรภาษาไทย และหลักทางภาษาศาสตร์แล้ว เราสามารถแบ่งตัวอักษรได้เป็น 5 กลุ่มใหญ่ๆคือ

1. พยัญชนะ ปัจจุบันมีใช้อยู่ 42 ตัว (ไม่นับ ข ค) เราสามารถแบ่งพยัญชนะออกเป็น 5 กลุ่มย่อย ๆ ดังนี้

1.1 พยัญชนะที่จะเป็นพยัญชนะต้นเสมอ ได้แก่

ฉ ผ ฝ ฮ

1.2 พยัญชนะที่ปกติจะเป็นพยัญชนะต้น ได้แก่

ห ภ ฆ ฟ ช ฑ

1.3 พยัญชนะที่เป็นได้ทั้งพยัญชนะและสระ ได้แก่

อ ว ร (ร ใช้ในรูป รร)

1.4 พยัญชนะที่ปกติจะเป็นตัวสะกด ได้แก่

ค ณ ญ ษ ฐ ฎ ฏ ฒ ฬ ฑ

1.5 พยัญชนะที่เป็นได้ทั้งตัวสะกดและพยัญชนะต้น ได้แก่

ก ข ค ง จ ช ด ต ถ ท ธ

น บ ป พ ม ย ล ส

2. <u>สระ</u> ที่ใช้อยู่ในปัจจุบันมี 17 ตัว ซึ่งสามารถแบ่งออกเป็นกลุ่มย่อย ได้ 5 กลุ่ม ดังนี้

2.1 สระ ที่ปกติจะเป็นตัวอักษรแรกของคำ ได้แก่

เ แ ไ ใ โ

2.2 สระที่ปกติจะเป็นตัวอักษรตัวสุดท้ายของคำ ได้แก่

ะ ำ

2.3 สระที่ปกติจะต้องการตัวสะกด ได้แก่

ั ็ ิ ึ

2.4 สระที่มี หรือ ไม่มีตัวสะกดก็ได้ ได้แก่

า ี ื

ุ ู

2.5 สระพิเศษ ที่ใช้เฉพาะในคำบางคำ ได้แก่

ฤ

3. <u>วรรณยุกต์</u> มี 4 ตัว ได้แก่ ่ ้ ๊ ๋

4. <u>สัญญลักษณ์พิเศษ</u> มี 21 ตัว ได้แก่

์ ฯ ๆ ( ) [_ { } " !

: ; | $ - . ? % / blank

5. <u>ตัวเลข</u> มี 10 ตัว ได้แก่ ตัวเลข 0-9

6.2 โครงสร้างโดยทั่วไปของคำในภาษาไทย

ตามหลักไวยากรณ์ของภาษาไทย สามารถแบ่งรูปแบบของคำได้ 7 รูปแบบดังนี้

1. [<ว>]
   <พ>[<พ>]<ส>[<ต>[<ต>]][<ก>] : จะ ถลา ถ้า กล้า บาน มฤตกานต์

2. [<ว>]
   <ส>
   <พ>[<พ>][<ต>[<ต>]][<ก>] : กิน หมั้น ที่ ฉันท์ คลี่ มือ

3. [<ว>]
   <ส>
   <พ>[<พ>][<ต>[<ต>]][<ก>] : จุ อุ๊ คุ้ม บูรณ์

4. [<ว>]
   <ส>
   <ส><พ>[<พ>][<ต>[<ต>]][<ก>] : ไพร แม่ โพยม โพธิ์

5. [<ว>]
   <ส>
   <ส><พ>[<พ>]<ต>[<ต>][<ก>] : เถิน เพลิน เปิ่น เป็น เซ็นต์

6. [<ว>]
   <ส><พ>[<พ>]<ว>[<ต>[<ต>]][<ก>] : เกือบ เกลียด เพลี้ย

7. [<ว>]
   <ส><พ>[<พ>]<ว>[<ว>]][<ก>] : เสาร์ เกล้า เข้า เกาะ เธอ

โดยที่ <พ> = พยัญชนะต้น <ส> = สระ
<ว> = วรรณยุกต์ <ต> = ตัวสะกด
<ก> = การันต์ [ ] = ให้เลือก อาจมีหรือไม่มีก็ได้

6.3 การสร้างกฎเกณฑ์ในการแบ่งคำ

จากการวิเคราะห์รูปแบบของคำ จึงได้สร้างกฎเกณฑ์ของการแบ่งคำเป็นข้อๆ โดยยึดหลักการทางภาษาศาสตร์และข้อมูลทางสถิติ แต่เนื่องจากภาษาไทยประกอบด้วยคำที่มีรูปแบบแตกต่างกันมากมาย ดังนั้นกฎที่ใช้ทุกกฎจึงต้องมีคำยกเว้นของกฎนั้นๆ

กฎเกณฑ์การแบ่งคำไทยที่สร้างขึ้นนี้ ได้รวบรวมเป็นหมวดหมู่ไว้แล้ว โดยจะยึดหลักการให้เป็นกฎที่มีความแน่นอน เพื่อจะใช้กับคอมพิวเตอร์ได้ โดยเฉพาะจะมีการกำหนดข้อยกเว้นต่างๆไว้ด้วย เพื่อให้มีความสมบูรณ์ของกฎเกณฑ์ ซึ่งจะมีหลักการดังนี้คือ

กฎข้อที่ 1 เครื่องหมายพิเศษ (Special Character) สามารถใช้แบ่งคำได้ โดยเราจะแบ่งเครื่องหมายพิเศษนี้ ออกเป็น 2 กลุ่ม ดังนี้คือ

- ประเภทที่เป็นวงเล็บเปิดและเครื่องหมายพิเศษบางตัว จะแบ่งคำหน้าตัวอักษรเหล่านั้น ได้แก่

  ([{ | /

  เช่น คำว่า "คำนาม(เอกพจน์)เป็นนามนับได้" จะตัดคำได้เป็น "คำนาม" และ "(เอกพจน์) เป็นนามนับได้"

- ประเภทที่เป็นวงเล็บปิดและเครื่องหมายพิเศษอื่นๆ จะแบ่งคำตรงตำแหน่งตัวอักษรนั้นๆ ได้แก่

)] } ! $ % - : ; ?

  เช่น คำว่า "(บรรทัด)ตรงกัน" จะตัดคำเป็น "(บรรทัด)" และ "ตรงกัน"

กฎข้อที่ 2 ตัวการันต์ (์) มักจะใช้เป็นตัวสุดท้ายของคำ เช่น ศิลป์ สันต์ เป็นต้น แต่ก็มีคำยกเว้นอยู่หลายคำ โดยมากมักจะเป็นคำที่มาจากภาษาอังกฤษ เช่น บอร์ด ฟิล์ม เป็นต้น ซึ่งคำเหล่านี้ มักจะนำหน้าการันต์ด้วยตัว ร และ ล เสมอ

กฎข้อที่ 3 ตัวอักษรสระอะ (ะ) และสระอำ (ำ) มักจะใช้เป็นตัวสุดท้ายของคำ ซึ่งก็มีข้อยกเว้น กรณีสระอะมีตัวอักษร ห์ ลงท้าย เช่น เคราะห์ เป็นต้น หรือกรณีสระอำ มีวรรณยุกต์ตามมาวรรณยุกต์ก็จะเป็นตัวสุดท้ายแทน เช่น ซ้ำ เป็นต้น

กฎข้อที่ 4 ตัวสระไม้ม้วน (ใ) จะใช้นำหน้าพยัญชนะเสมอ จึงจะเป็นตัวอักษรแรกสุดของคำ (คำในภาษาไทยที่ใช้ ใ มี 20 คำ เท่านั้น)

กฎข้อที่ 5 ตัวสระไม้หันอากาศ (ั) สระอิ (ิ) สระอี (ี) และ สระอึ (ึ) โดยปกติมักจะต้องการตัวสะกด 1 ตัว เช่น กิน ตัด ยืด แต่มีคำยกเว้นหลายคำ เช่น นัยน์ เกิน เรือง เป็นต้น

กฎข้อที่ 6 ตัวสระเอ (เ) สระแอ (แ) สระโอ (โ) และ สระไม้มลาย (ไ) ปกติจะใช้นำหน้าพยัญชนะ แต่ก็จะมีคำยกเว้นอยู่หลายคำ เช่น มเหสี สแลง อโหสิ สไบ เป็นต้น

กฎข้อที่ 7 ตัวพยัญชนะ ฉ ผ ฝ อ จะใช้เป็นพยัญชนะต้นนำหน้าเสมอ เช่น ฉกรรจ์ ผไท ฝรั่ง อ้อ เป็นต้น แต่อาจจะมีตัวสระในกฎข้อที่ 4, 6 นำหน้าได้ เช่น เฉลียง เผย โอ้ง เป็นต้น

กฎข้อที่ 8 ตัวสระอุ (ุ) และ สระอู (ู) มักจะใช้ไว้ใต้พยัญชนะตัวแรก หรือพยัญชนะตัวที่สองของคำ เช่น คุณ กุล ปลุก สนุก ขูด มูก กรูด อณู เป็นต้น แต่อาจจะมีคำที่สระ ไปอยู่ใต้พยัญชนะตัวอื่นๆ (ไม่ใช่พยัญชนะตัวแรกหรือตัวที่สอง) เช่น เหตุ ธาตุ เรณู เมนู ไอศูรย์ เป็นต้น

กฎข้อที่ 9 ตัวสระอา (า) โดยปกติจะต้องมีพยัญชนะนำหน้าอย่างน้อย 1 ตัว เสมอ โดยจะพิจารณาเป็น 2 กรณี คือกรณีที่ในคำไม่มีการใช้วรรณยุกต์เลย เช่น ปากกา นาน สบาย สตางค์ และกรณีที่ในคำมีการใช้วรรณยุกต์ร่วมอยู่ด้วย เช่น กร๊าฟ ฝ่าย หม้าย เป็นต้น โดยในแต่ละกรณีจะแบ่งคำหน้าหรือหลังสระก็ได้ แล้วแต่ความเหมาะสมของรูปแบบของคำนั้นๆ

กฎข้อที่ 10 ตัวอักษร อ ที่ใช้ร่วมกับวรรณยุกต์ จะพิจารณาจากตัวอักษรที่อยู่หน้าและหลังตัวอักษร อ โดยจะดูว่าพยัญชนะต้นแต่ละตัวนั้น เมื่อใช้กับ อ แล้ว จะมีตัวสะกดเป็นตัวใดได้บ้าง เช่น พยัญชนะต้นเป็นตัว ก สำหรับวรรณยุกต์ (่) แล้ว จะมีตัวสะกดเป็นตัวใดได้บ้าง เช่น พยัญชนะต้นเป็นตัว ก สำหรับวรรณยุกต์แล้ว จะมีตัวสะกดเพียงตัวเดียวคือ น (ก่อน) ส่วนวรรณยุกต์ (้) จะมีตัวสะกด คือ น ง ย (ก้อน, ก้อง, ก้อย) เป็นต้น

กฎข้อที่ 11 ตัวอักษรไม้ไต่คู้ (ั) จะเป็นตัวที่เปลี่ยนรูปมาจากสระ เ-าะ เ-ะ แ-ะ ที่มีตัวสะกด ดังนั้นรูปแบบที่ใช้จะเป็น ั-อ- เ-็-- แ-็-- ทุกรูปแบบจะมีตัวสะกด 1 ตัว ยกเว้นถ้ามีตัวการันต์

กฎข้อที่ 12 สระผสม เ-ีย และ เ-ือ ที่จะใช้ร่วมกับวรรณยุกต์ต่างๆ โดยจะพิจารณาว่าแต่ละรูปแบบจะมีตัวสะกดหรือไม่และถ้ามีตัวสะกด ก็จะใช้ตัวใดได้บ้าง เพื่อจะพิจารณาตำแหน่งแบ่งคำก่อนหน้าหรือหลังสระนั้น

กฎข้อที่ 13 ตัวอักษร ฤ โดยปกติจะใช้อยู่ถัดไป จากตัวพยัญชนะต้นนำหน้า เช่น กฤษณะ หฤทัย คฤหัสถ์ พฤกษ์ เป็นต้น แต่ก็จะมีข้อยกเว้นที่ใช้ตัวอักษร ฤ เป็นพยัญชนะต้นนำหน้าได้ เช่น ฤดู ฤดี ฤชา ฤกษ์ ฤทธิ์ ฤทัย ฤาษี เป็นต้น

กฎข้อที่ 14 ตัวอักษร ห มักจะใช้เป็นพยัญชนะนำหน้าเสมอ แต่ก็มีคำยกเว้น เช่น สหัส มหา คหบดี มหกรรม มหรสพ มหัศ มหิ พรหม เคราะห์ เป็นต้น นอกจากนั้นจะเป็นคำที่มาจากต่างประเทศ เช่น จอห์น โอห์ม เป็นต้น หรือมีคำนำหน้าเป็นสระ เช่น เหา แห่ง เป็นต้น

กฎข้อที่ 15 ตัวอักษร ว จะต้องมีตัวสะกดอย่างน้อย 1 ตัว เมื่อใช้อยู่ต่อจากวรรณยุกต์ เช่น ม้วน ล้วน ม่วง เป็นต้น และอาจจะใช้ในรูปของสระอัว (ั-ว) เช่น ตัว มัว ขั้ว เป็นต้น

กฎข้อที่ 16 ตัวอักษร ร โดยปกติจะใช้ในรูปแบบของสระ -รร เช่น วรรณ จรรยา บรรจง เป็นต้น

กฎข้อที่ 17 เป็นกฎของสระลดรูป ซึ่งจะมีตัวสะกดเป็นตัวพยัญชนะต่อไปนี้ คือ ก ง ค น ม บ เช่น คน ชก กด เป็นต้น

กฎข้อที่ 18 ตัวอักษร ศ ณ ญ ษ ฐ ฎ ฏ ฌ ฬ ฆ มักจะใช้เป็นตัวสะกดเสมอ เช่น กิฬา คณิกา อุษา อัศจรรย์ หญิง ชฎา ปฏัก แต่ก็มีข้อยกเว้นที่จะใช้เป็นพยัญชนะต้นนำหน้าคำได้ เช่น ศุลี ฐาน ฎีกา ญวน เป็นต้น

6.4 หลักการของการแบ่งคำภาษาไทย

จากกฎเกณฑ์ของการแบ่งคำ เราสามารถนำมาใช้เป็นหลักการที่แน่นอน เพื่อเขียนโปรแกรมให้คอมพิวเตอร์ทำงานได้ตามต้องการ โดยในที่นี้จะอยู่ภายใต้โมดูลที่ชื่อว่า "FINDCUT"

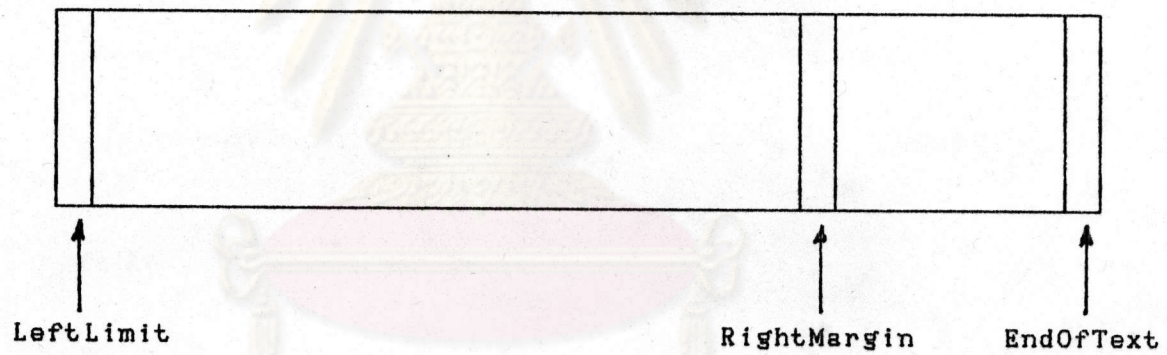
การเรียกใช้โมดูลนี้ จะต้องมีเนื้อที่บัฟเฟอร์ ซึ่งใช้เก็บข้อความที่จะถูกตัดคำและกำหนดค่าของพอยน์เตอร์ที่สำคัญอีก 3 ตัว คือ

LeftLimit - เป็นพอยน์เตอร์ที่จะชี้ไปยังตัวซ้ายสุด ของเนื้อที่บัฟเฟอร์

RightMargin - เป็นพอยน์เตอร์ที่จะชี้ไปยังตัวขวาสุด ของเนื้อที่บัฟเฟอร์ (ตัวกั้นหลังขวา) ซึ่งเป็นจุดเริ่มต้นให้มีการเริ่มต้นแบ่งคำ

EndOfText - เป็นพอยน์เตอร์ที่จะชี้ไปยังตัวท้ายสุด ของเนื้อที่บัฟเฟอร์

ดังแสดงการใช้พอยน์เตอร์ดังรูปที่ 6.1



รูปภาพที่ 6.1 แสดงพอยน์เตอร์ที่จะใช้ในการตัดคำ

การทำงานของโมดูลนี้ จะพยายามหาจุดแบ่งคำที่อยู่ระหว่าง LeftLimit กับ RightMargin โดยจะให้อยู่ใกล้ RightMargin มากที่สุด ซึ่งจะให้ผลลัพธ์เป็นพอยน์เตอร์ชี้ไปยังตัวอักษรที่สามารถแบ่งคำได้ หลักการทำงานของโมดูลนั้น เริ่มแรกจะพยายามแบ่งที่ตัว RightMargin ก่อน โดยจะใช้การทำงานในโมดูลย่อยๆตามค่าของตัวอักษรนั้นๆ ถ้าแบ่งคำยังไม่ได้ ก็จะถอยมาทางซ้ายเรื่อยๆ จนกว่าจะแบ่งคำได้ หรือจนกระทั่งถึง LeftLimit

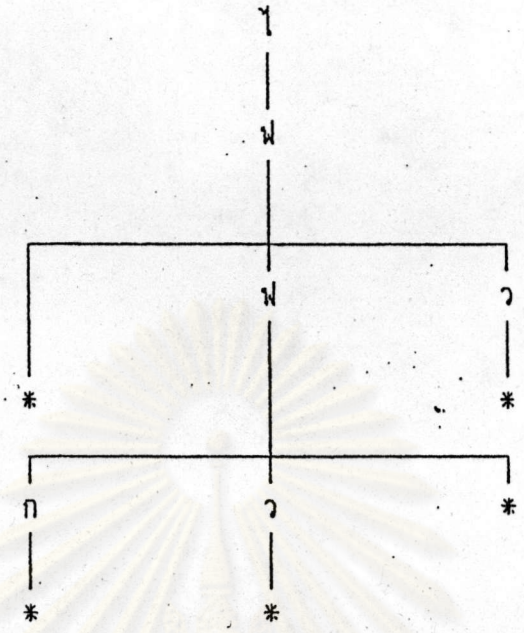
การแบ่งคำนั้นจะทำตามกฎพื้นฐานก่อน จากนั้นจึงดูว่าเป็นคำยกเว้นหรือไม่ ในการค้นหาคำยกเว้นที่มีอยู่มากมายในแต่ละกฎนั้น จะใช้การค้นหาแบบไบนารี่ทรี ซึ่งจะทำได้เร็วกว่าการค้นหาแบบเรียงลำดับ ตัวอย่างเช่น กฎที่เป็นไปได้ของไม้มลัย (ไ) ที่เป็นไปได้ จะมีดังนี้คือ

ไ<พ>*	เช่น ไข ไป ไว ไฟ
ไ<พ><ว>*	เช่น ไก่ ไข่ ไว้ ไม่
ไ<พ><พ>*	เช่น ไกล ไหน ไหม ไตร
ไ<พ><พ><ว>*	เช่น ไขว่ ไหว้ ไตร่ ไหม้
ไ<พ><พ><ก>*	เช่น ไมล์ ไนต์ ไฟล์ ไกด์

เมื่อนำกฎของตัวไม้มลัย (ไ) มาเขียนเป็นเส้นทางการค้นหาแบบไบนารี่ทรี จะได้ผลดังรูปที่ 6.2 โดยเมื่อพบตัวไม้มลัยแล้วจะไปค้นหาในแต่ละเส้นทางที่เป็นไปได้ก่อน ถ้าพบข้อมูลที่ต้องการแล้ว ก็จะค้นหาลึกลงไปในแต่ละเส้นทางนั้น ซึ่งถ้าพบว่าเป็นคำยกเว้น ก็จะตัดคำตามคำยกเว้นนั้น แต่ถ้าไม่พบก็จะตัดที่ตัวก่อนหน้าตัวไม้มลัย ตามกฎพื้นฐาน

อนึ่งการทำงานจะต้องมีการตรวจสอบชนิดของตัวอักษรบ่อยครั้ง เพื่อช่วยในการตรวจสอบชนิดของตัวอักษรที่เข้ามาให้ได้รวดเร็ว จะมีการใช้ข้อมูลชุดหนึ่งขนาด 16 บิท เป็นตัวกำหนดชนิดของตัวอักษรโดยแต่ละบิท จะมีความหมาย ดังนี้

bit 0	แทน ตัวเลขไทย	bit 6	แทน พยัญชนะที่เป็นพยัญชนะต้นเสมอ
bit 1	แทน เครื่องหมายพิเศษ	bit 7	แทน พยัญชนะ
bit 2	แทน การันต์	bit 8	แทน วงเล็บปิดและเครื่องหมายพิเศษอื่น
bit 3	แทน วรรณยุกต์	bit 9	แทน วงเล็บเปิด
bit 4	แทน สระตาม	bit 10 - 15	ยังไม่ได้ใช้
bit 5	แทน สระนำ		



รูปภาพที่ 6.2 แสดงเส้นทางการค้นหาแบบไบนารี่ทรี ของกฎตัวไม้มลัย

ซึ่งจะทำให้ได้ค่าของข้อมูลทั้งหมด 11 แบบ ที่จะนำไปเปรียบเทียบกับค่าของตัวอักษรได้ ดังนี้คือ

0000001000000000 (ฐานสอง) หรือ 0x0100 (ฐานสิบหก) แทน ตัววงเล็บเปิด

0000000100000000 (ฐานสอง) หรือ 0x0200 (ฐานสิบหก) แทน ตัววงเล็บปิด

0000000010000000 (ฐานสอง) หรือ 0x80 (ฐานสิบหก) แทน ตัวพยัญชนะ

0000000001000000 (ฐานสอง) หรือ 0x40 (ฐานสิบหก) แทน ตัวพยัญชนะต้น

0000000000110000 (ฐานสอง) หรือ 0x30 (ฐานสิบหก) แทน ตัวสระ

0000000000100000 (ฐานสอง) หรือ 0x20 (ฐานสิบหก) แทน ตัวสระนำ

0000000000010000 (ฐานสอง) หรือ 0x10 (ฐานสิบหก) แทน ตัวสระตาม

0000000000001000 (ฐานสอง) หรือ 0x08 (ฐานสิบหก) แทน ตัววรรณยุกต์

0000000000000100 (ฐานสอง) หรือ 0x04 (ฐานสิบหก) แทน ตัวการันต์

0000000000000010 (ฐานสอง) หรือ 0x02 (ฐานสิบหก) แทน ตัวเครื่องหมายพิเศษ

0000000000000001 (ฐานสอง) หรือ 0x01 (ฐานสิบหก) แทน ตัวเลขไทย

ตัวอย่างเช่น ก เป็นตัวอักษรมีค่าเป็น 0x00A1 (เลขฐานสิบหก) หรือ 0000000010100001 ถ้านำไปกระทำทางลอจิกด้วยการ AND กับค่า 0000000010000000 จะได้ 0000000010000000 เหมือนเดิม ซึ่งมีค่าไม่เป็นศูนย์ แสดงว่า ก เป็นตัวพยัญชนะจริงแต่ในขณะเดียวกันถ้าไปกระทำทางลอจิกด้วยการ AND กับค่า 0000000001000000 จะได้เป็นศูนย์แสดงว่า ก ไม่ใช่ตัวพยัญชนะต้นเป็นเพียงตัวพยัญชนะธรรมดาเท่านั้น

หลักการแบ่งคำภาษาไทย จะมีขั้นตอนการทำงานได้ดังนี้คือ

1. กำหนดให้พอยน์เตอร์ Indx เป็นค่า RightMargin (ช่วงเว้นขอบขวา)
2. ตรวจสอบค่าของ Indx ว่ายังมีค่าเกินกว่า LeftLimit หรือไม่

- ถ้ายังมีค่าเกินกว่า LeftLimit ขึ้นไป แล้วจะทำงานตามขั้นตอนดังนี้คือ

2.1 ไปทำงานในโมดูล "ThaiEngCutRtn" เพื่อจะใช้แบ่งคำภาษาไทยในตำแหน่งก่อนหน้า ซึ่งพอยน์เตอร์ Indx ชี้อยู่ โดยจะตรวจสอบว่าสามารถแบ่งคำตามที่ต้องการหรือไม่ ถ้าแบ่งคำได้ก็จะให้ตำแหน่งก่อน

หน้านั้น แต่ถ้าแบ่งคำไม่ได้จะทำงานถัดไป

2.2 ทำงานในโมดูล "FuncPtr" โดยใช้ตัวอักษรในตำแหน่งที่พอยน์เตอร์ indx ชี้อยู่ นำไปค้นหาในตารางฟังก์ชั่น เพื่อให้ได้ชื่อของโมดูลที่จะไปทำงานในโมดูลนั้นๆ ซึ่งในแต่ละโมดูลจะตรวจสอบว่าตัวอักษรในตำแหน่งนั้น สามารถแบ่งคำตามกฎเกณฑ์ต่างๆข้างต้นได้จริงหรือไม่ ถ้าเป็นจริงแล้ว จะให้ผลลัพธ์เป็นตำแหน่งของคำที่จะใช้แบ่งได้ แต่ถ้าไม่เป็นจริงแล้ว แสดงตัวอักษรในตำแหน่งนั้นแบ่งคำไม่ได้ ก็จะลดค่าของ Indx ลงอีก 1 เพื่อที่จะขยับไปยังตัวอักษรทางซ้ายถัดไปอีกแล้วจะกลับไปทำงานในหัวข้อ 2 อีกครั้ง

- ถ้ามีค่าน้อยกว่า LeftLimit แล้ว แสดงว่าไม่สามารถหาตำแหน่งที่จะแบ่งคำได้เลย จะให้ผลลัพธ์ของตำแหน่งที่จะแบ่งคำเป็น RightMargin แทนได้

สำหรับขั้นตอนของการทำงาน ได้เขียนเป็นผังงานไว้ในภาพที่ 6.3 แล้ว

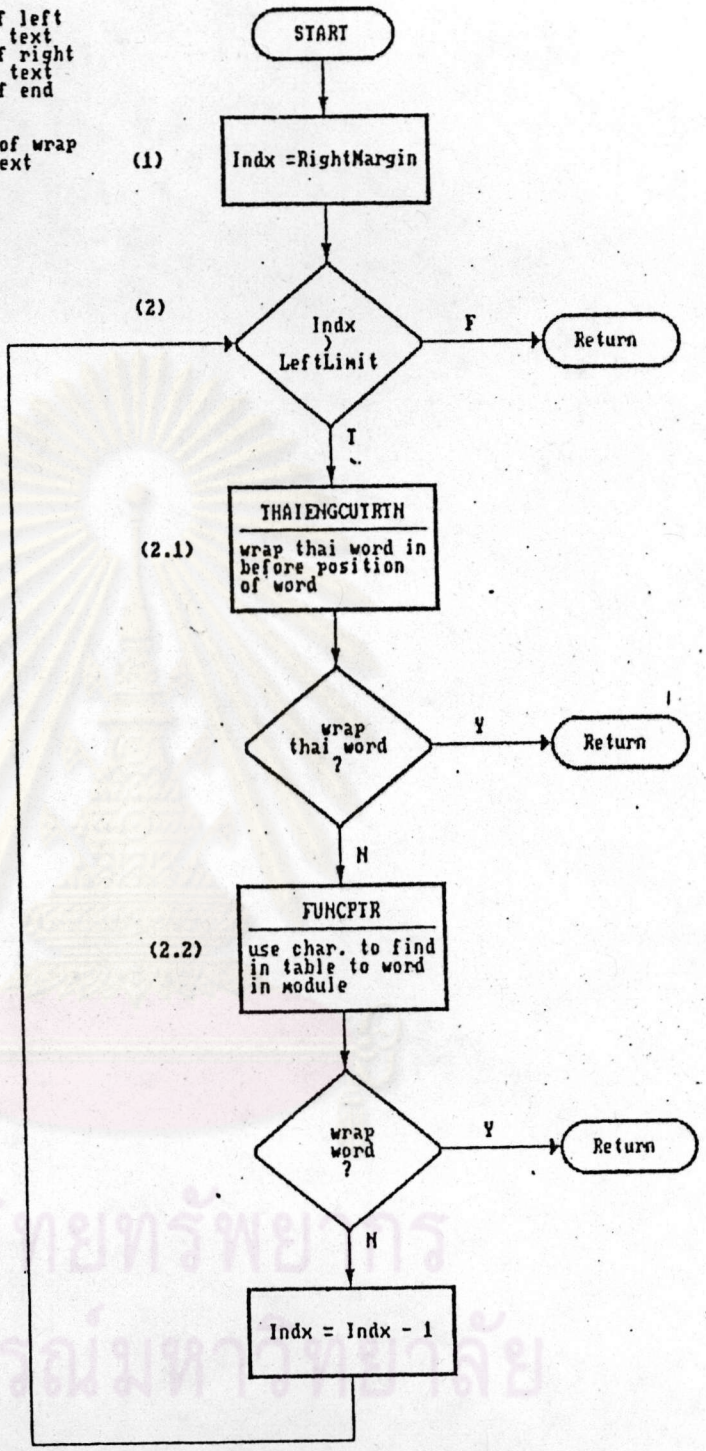
ส่วนการทำงานในโมดูล "FuncPtr" ที่ใช้ตัวอักษรในตำแหน่งต้องการไปค้นหาในตารางฟังก์ชั่นเพื่อจะนำชื่อของโมดูลมาทำงานนั้น สามารถเขียนเป็นตารางความสัมพันธ์ระหว่างตัวอักษรกับชื่อโมดูลได้ดังนี้คือ

	ตัวอักษร	คำอธิบาย	ชื่อของโมดูล
1.	({ [/ !	เครื่องหมายวงเล็บเปิด	FcPuncChar
2.	) }] ! $ % - : ; ?	เครื่องหมายวงเล็บปิดและเครื่องหมายพิเศษ	PuncChar
3.	ฉ ผ ฝ ฮ	พยัญชนะต้น	LeadConRtn
4.	์	การันต์	KaranRtn
5.	ร	ตัวอักษร ร	RoreKeoRtn
6.	ฤ	ตัวอักษร ฤ	RoreRURtn
7.	ว	ตัวอักษร ว	WoreWaanRtn

Flow of FINDCUT module

INPUT : LeftLimit - pointer of left margin of text
RightMargin - pointer of right margin of text
EndOfText - pointer of end of text

OUTPUT : cuthere - position of wrap word in text



รูปภาพที่ 6.3 ผังงานแสดงขั้นตอนการทำงานของโมดูล FINDCUT

8.	ห	ตัวอักษร ห	HoreHeebRtn
9.	อ	ตัวอักษร อ	OrAngRtn
10.	ฯ	ตัวอักษร ฯ	PaiYarnnoyRtn
11.	ะ	สระอะ	SaraAhRtn
12.	ั	ไม้หันอากาศ	HunAkadRtn
13.	า	สระอา	SaraRRtn
14.	ำ	สระอำ	SaraUmRtn
15.	ิ	สระอิ	SaraIeRtn
16.	ี	สระอี	SaraERtn
17.	ึ	สระอึ	SaraUeRtn
18.	ื	สระอือ	SaraUeeRtn
19.	ุ	สระอุ	SaraURtn
20.	ู	สระอู	SaraUURtn
21.	เ	สระเอ	SaraARtn
22.	แ	สระแอ	SaraAirRtn
23.	โ	สระโอ	SaraORtn
24.	ใ	สระไม้ม้วน	MaiMuanRtn
25.	ไ	สระไม้มลัย	MaiMalaiRtn
26.	ๆ	ตัวอักษรไม้ยมก	MaiYamokRtn
27.	็	สระไม้ไต่คู้	MaiTaiKuuRtn
28.	,	เครื่องหมายคอมม่า	TermChar
29.	ตัวอักษรที่มีค่าระหว่าง 0x01 ถึง 0x20	เครื่องหมายรหัสควบคุม	CntlChar
30.	ตัวอักษรอื่นๆ	ตัวอักษรอื่นๆที่นอกเหนือจากที่กล่าวถึงทั้งหมด	dummy

ตัวอย่างเช่น ถ้าตัวอักษร [เมื่อไปทำงานในโมดูล "FuncPtr" แล้วจะนำตัวอักษรไปค้นหาในตารางฟังก์ชั่น ซึ่งจะทำให้ได้ชื่อโมดูลเป็น FCPuncChar ที่จะไปทำงานต่อไป หรือ ถ้าตัวอักษร อ เมื่อไปทำงานในโมดูล "FuncPtr" แล้ว จะนำตัวอักษรไปค้นหาในตารางฟังก์ชั่นซึ่งจะทำให้ได้ชื่อโมดูล เป็น OrAngRtn ที่จะไปทำงานต่อไปได้ หรือถ้าตัวอักษรเป็นตัวอักษรภาษาอังกฤษเมื่อไปทำงานในโมดูล "FuncPtr" แล้ว จะนำตัวอักษรไปค้นหาในตารางฟังก์ชั่นซึ่งจะทำให้ได้ชื่อโมดูลเป็น dummy

6.5 การแบ่งคำในโปรแกรมประมวลผลคำภาษาไทย

จากหลักการของการแบ่งคำภาษาไทย จะนำมาประยุกต์ใช้กับการประมวลผลคำภาษาไทย เพื่อใช้จัดให้ข้อความด้านขวาตรงกันตลอด ทำให้การทำงานของการประมวลคำภาษาไทยมีประสิทธิภาพดีขึ้น

การแบ่งคำในโปรแกรมประมวลผลคำภาษาไทยจะใช้ข้อความในบรรทัดที่ใช้งานอยู่ขณะนั้น ซึ่งจะกำหนดพอยน์เตอร์ 3 ตัว คือ พอยน์เตอร์ที่ชี้ไปยังตำแหน่งขอบซ้ายสุดของบรรทัด พอยน์เตอร์ที่ชี้ไปยังตำแหน่งขอบขวาสุดของบรรทัด และพอยน์เตอร์ที่ชี้ไปยังตำแหน่งท้ายสุดของบรรทัด แล้วไปทำงานในโมดูล "FINDCUT" ซึ่งจะได้ผลลัพธ์เป็นตำแหน่งของคำที่จะถูกตัดคำได้ โดยจะทำการแบ่งข้อความตั้งแต่ตำแหน่งผลลัพธ์นั้น ไปขึ้นบรรทัดใหม่เอง และปรับข้อความในบรรทัดที่ถูกแบ่งคำให้ชิดขวาตรงกัน

การทำงานของการแบ่งคำในโปรแกรมประมวลผลคำภาษาไทย จะอยู่ภายใต้โมดูล "autowrap" โดยบรรทัดที่ใช้งานอยู่ จะถูกชี้ด้วยพอยน์เตอร์ curline ซึ่งจะมีขั้นตอนการทำงานได้ดังนี้คือ

1. กำหนดพอยน์เตอร์ temp2 ให้ชี้ไปตำแหน่งขอบซ้ายสุดของบรรทัดที่ใช้อยู่ขณะนั้น (left margin)
2. กำหนดพอยน์เตอร์ temp1 ให้ชี้ไปตำแหน่งขอบขวาสุดของบรรทัดที่ใช้อยู่ขณะนั้น (Right margin)
3. กำหนดพอยน์เตอร์ temp3 ให้ชี้ไปตำแหน่งท้ายสุดของบรรทัดที่ใช้อยู่ขณะนั้น (End of Text)

4. ทำงานในโมดูล "FINDCUT" เพื่อให้หาตำแหน่งที่จะแบ่งคำ เก็บไว้ในพอยน์เตอร์ cuthere

5. สร้างบรรทัดใหม่เพื่อใช้เก็บข้อความที่จะถูกแบ่งคำหลังตำแหน่ง cuthere มาแทรกไว้ในระหว่างบรรทัดที่ใช้งานอยู่ ซึ่งจะมีขั้นตอนดังนี้คือ

   5.1 กำหนดให้พอยน์เตอร์ templine ชี้ไปโหนดพอยน์เตอร์ที่สร้างใหม่

   5.2 สร้างบัฟเฟอร์โหนดใหม่ โดยให้ถูกชี้ด้วยโหนดพอยน์เตอร์นั้น และทำการคัดลอกข้อมูลจากบรรทัดที่ใช้งานอยู่ตั้งแต่ตำแหน่งที่ cuthere ไปจนหมดบรรทัด มาเก็บไว้ในบัฟเฟอร์โหนด

   5.3 กำหนดให้พอยน์เตอร์ temp3 ชี้ไปโหนดพอยน์เตอร์ที่สร้างใหม่

   5.4 สร้างบัฟเฟอร์โหนดใหม่ โดยให้ถูกชี้ด้วยโหนดพอยน์เตอร์นั้น และทำการคัดลอกข้อมูลจากบรรทัดที่ใช้งานอยู่ ตั้งแต่ตำแหน่งแรกไปจนถึงตำแหน่ง cuthere มาเก็บไว้ในบัฟเฟอร์โหนด

   5.5 คัดลอกข้อความในบัฟเฟอร์โหนดของ temp3 ไปเก็บไว้ในบรรทัดที่ใช้งานอยู่

   5.6 นำบรรทัดที่ templine ชี้อยู่ไปเชื่อมต่อกับบรรทัดที่ใช้งานอยู่

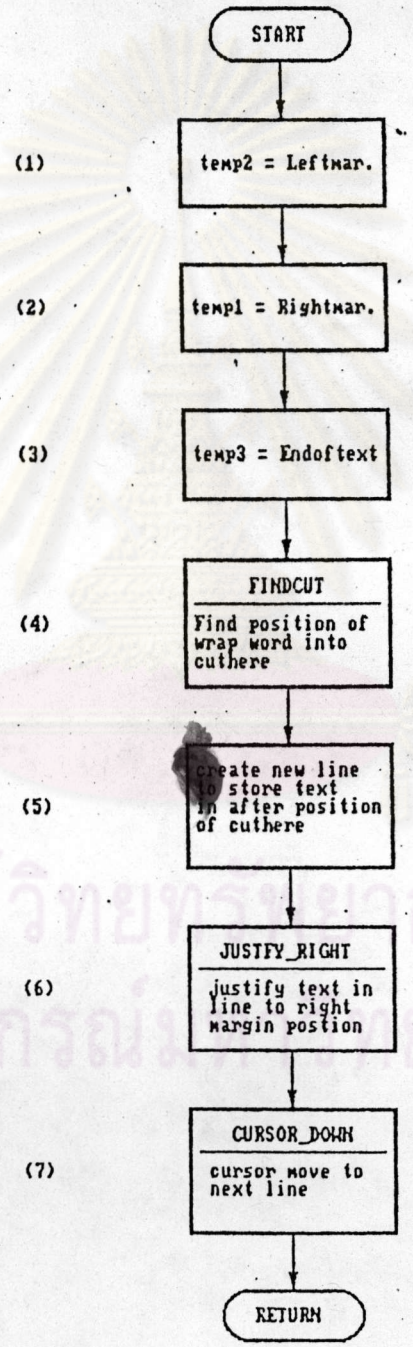
6. ทำงานในโมดูล "justify_right" เพื่อทำการปรับข้อความในบัฟเฟอร์ทำงานระดับกลางให้ได้มีช่วงขอบขวาตรงกันโดยการแทรกช่องว่างระหว่างตัวอักษรเข้าไป

7. ทำงานในโมดูล "cursor_down" เพื่อจะเลื่อนเคอร์เซอร์ให้ลงมาอีกบรรทัดหนึ่ง

สำหรับขั้นตอนของการทำงาน ได้เขียนเป็นผังงานไว้ในภาพที่ 6.4 แล้ว

นอกจากนั้นแล้ว การทำงานของการแบ่งคำในโปรแกรมประมวลผลคำภาษาไทยอาจจะอยู่ภายใต้โมดูล "reform" ซึ่งการทำงานจะคล้ายๆกับโมดูล "autowrap" แต่จะทำงานทีละย่อหน้า โดยจะนำบรรทัดข้อความที่ยาวเกินช่วงเว้นขอบขวา ไปหาตำแหน่งที่ใช้แบ่งข้อความได้ ซึ่งจะทำการตัดข้อความบรรทัดตั้งแต่ในตำแหน่งนั้นไปขึ้นบรรทัดใหม่เอง และปรับข้อความในบรรทัดที่ถูกแบ่งคำให้ชิดขวาตรงกันซึ่งจะทำทีละย่อหน้า

Flow of AUTOWRAP module



รูปภาพที่ 6.4 ผังงานแสดงขั้นตอนการทำงานของโมดูล autowrap

6.6 สรุปโมดูลย่อยที่ใช้ในการแบ่งคำภาษาไทยในโปรแกรมซียูไรท์เตอร์

จะมีโมดูลย่อยที่สำคัญ ต่อการใช้งานดังนี้คือ

FINDCUT - จะใช้สำหรับหาตำแหน่งที่จะแบ่งข้อความให้อย่างเหมาะสมของบรรทัดที่ใช้งานอยู่ โดยจะกำหนดค่า 3 อย่าง คือ ตำแหน่งขอบซ้ายสุดของบรรทัด ตำแหน่งขอบขวาสุดของบรรทัด และ ตำแหน่งท้ายสุดของบรรทัด

justify_right - ปรับข้อความในบรรทัดที่ใช้งานอยู่ ให้มีช่วงเว้นขอบขวาตรงกัน โดยจะใส่ช่องว่างระหว่างคำให้เพิ่มขึ้น

autowrap - ใช้สำหรับแบ่งคำในการประมวลผลภาษาไทย ของบรรทัดที่ใช้งานอยู่ โดยจะทำทีละหนึ่งบรรทัด

reform - ใช้สำหรับแบ่งคำในการประมวลผลภาษาไทยโดยจะทำทีละหนึ่งย่อหน้า

nstrcmp - ใช้เปรียบเทียบข้อความ 2 ตัว ว่าเหมือนกันหรือไม่

ThaiEngCutRtn - เพื่อจะใช้แบ่งคำภาษาไทยในตำแหน่งก่อนหน้า ที่ใช้งานอยู่ โดยจะตรวจสอบว่าสามารถแบ่งคำตามที่ต้องการได้หรือไม่

FuncPtr - ใช้ตัวอักษรในตำแหน่งที่ใช้แบ่งคำไปค้นหาในตารางฟังก์ชั่นเพื่อค้นหาชื่อของโมดูล ซึ่งแต่ละโมดูลจะเป็นกฎเกณฑ์การแบ่งคำที่จะนำไปทำงานได้

FcPuncChar - กฎเกณฑ์การแบ่งคำ เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำเป็นเครื่องหมายวงเล็บเปิด ได้แก่ ({ [/ ¦

PuncChar - กฎเกณฑ์การแบ่งคำ เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำเป็นเครื่องหมายวงเล็บปิดและเครื่องหมายพิเศษ ได้แก่

) }] ¦ $ % - : ; ?

LeadConRtn - กฎเกณฑ์การแบ่งคำ เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำเป็นพยัญชนะต้นนำหน้าเสมอ ได้แก่ ฉ ผ ฝ ฮ

KaranRtn	-	กฎเกณฑ์การแบ่งคำ เป็นตัวอักษรการันต์	เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำ
RoreReoRtn	-	กฎเกณฑ์การแบ่งคำ เป็นตัวอักษร ร	เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำ
RoreRuRtn	-	กฎเกณฑ์การแบ่งคำ เป็นตัวอักษร ฤ	เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำ
WoreWaanRtn	-	กฎเกณฑ์การแบ่งคำ เป็นตัวอักษร ว	เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำ
HorHeebRtn	-	กฎเกณฑ์การแบ่งคำ เป็นตัวอักษร ห	เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำ
OrAngRtn	-	กฎเกณฑ์การแบ่งคำ เป็นตัวอักษร อ	เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำ
PaiYarnnoyRtn	-	กฎเกณฑ์การแบ่งคำ เป็นตัวอักษร ฯ	เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำ
SaraAhRtn	-	กฎเกณฑ์การแบ่งคำ เป็นตัวอักษรสระอะ	เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำ
HunAkadRtn	-	กฎเกณฑ์การแบ่งคำ เป็นตัวอักษรไม้หันอากาศ	เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำ
SaraRRtn	-	กฎเกณฑ์การแบ่งคำ เป็นตัวอักษรสระอา	เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำ
SaraUmRtn	-	กฎเกณฑ์การแบ่งคำ เป็นตัวอักษรสระอำ	เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำ
SaraIeRtn	-	กฎเกณฑ์การแบ่งคำ เป็นตัวอักษรสระอิ	เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำ
SaraERtn	-	กฎเกณฑ์การแบ่งคำ เป็นตัวอักษรสระอี	เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำ

SaraUeRtn - กฎเกณฑ์การแบ่งคำ เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำเป็นตัวอักษรสระอิ

SaraUeeRtn - กฎเกณฑ์การแบ่งคำ เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำเป็นตัวอักษรสระอือ

SaraURtn - กฎเกณฑ์การแบ่งคำ เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำเป็นตัวอักษรสระอุ

SaraUURtn - กฎเกณฑ์การแบ่งคำ เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำเป็นตัวอักษรสระอู

SaraARtn - กฎเกณฑ์การแบ่งคำ เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำเป็นตัวอักษรสระเอ

SaraAirRtn - กฎเกณฑ์การแบ่งคำ เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำเป็นตัวอักษรสระแอ

SaraORtn - กฎเกณฑ์การแบ่งคำ เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำเป็นตัวอักษรสระโอ

MaiMuanRtn - กฎเกณฑ์การแบ่งคำ เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำเป็นตัวอักษรไม้ม้วน

MaiMalaiRtn - กฎเกณฑ์การแบ่งคำ เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำเป็นตัวอักษรไม้มลัย

MaiYamokRtn - กฎเกณฑ์การแบ่งคำ เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำเป็นตัวอักษรไม้ยมก

MaiTaiKuuRtn - กฎเกณฑ์การแบ่งคำ เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำเป็นตัวอักษรไม้ไต่คู้

SaraAEEkRtn - กฎเกณฑ์การแบ่งคำ เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำเป็นตัวอักษรสระเอียไม้เอก

SaraAEToeRtn - กฎเกณฑ์การแบ่งคำ เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำเป็นตัวอักษรสระเอียไม้โท

SaraAETriRtn - กฎเกณฑ์การแบ่งคำ เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำ เป็นตัวอักษรสระเอียไม้ตรี

SaraAEJatRtn - กฎเกณฑ์การแบ่งคำ เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำ เป็นตัวอักษรสระเอียจัตวา

SaraAERtn - กฎเกณฑ์การแบ่งคำ เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำ เป็นตัวอักษรสระเอีย

SaraErEkRtn - กฎเกณฑ์การแบ่งคำ เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำ เป็นตัวอักษรสระเออไม้เอก

SaraErToeRtn - กฎเกณฑ์การแบ่งคำ เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำ เป็นตัวอักษรสระเออไม้โท

SaraErRtn - กฎเกณฑ์การแบ่งคำ เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำ เป็นตัวอักษรสระเออ

TermChar - กฎเกณฑ์การแบ่งคำ เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำ เป็นตัวอักษรเครื่องหมายคอมม่า

CntlChar - กฎเกณฑ์การแบ่งคำ เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำ เป็นตัวอักษรรหัสควบคุม

dummy - กฎเกณฑ์การแบ่งคำ เมื่อตัวอักษรในตำแหน่งที่ใช้แบ่งคำ เป็นตัวอักษรอื่นๆ เช่น ตัวอักษรภาษาอังกฤษ