บทที่ 3

การออกแบบและพัฒนาโปรแกรมสำหรับแสดงผลตัวอักษรไทยบนระบบเท็กซ์

ในบทที่แล้วได้ศึกษาถึงการทำงานของเท็กซ์ การทำงานของเมตาฟอนต์ และการตัดคำไปแล้วในบทนี้จะได้นำความรู้ที่ได้จากการศึกษาดังกล่าวมาพัฒนาให้เป็นโปรแกรมไทยเท็กซ์

ในที่นี้เราจะแยกงานทั้งหมดออกเป็น 3 ส่วนดังนี้

1. <u>การสร้างฟอนต์ภาษาไทยด้วยเมตาฟอนต์</u>

ขั้นตอนนี้จะทำการเลือกฟอนต์ชนิดหนึ่งขึ้นมาเป็นแบบอย่าง ซึ่งในที่นี้เราจะนำมาจากโปรแกรม Microsoft Word ที่ใช้อยู่บนวินโดว์ทั่วไป นำฟอนต์นั้นมาขยายขนาดแล้วนำมาตัดตามขอบของตัวอักษรแต่ละตัว จากนั้นเราจะใช้กระดาษกราฟเป็นแกนอ้างอิงซึ่งวิธีการนี้จะใกล้เคียงกันมากกับระบบของเมตาฟอนต์ที่เสมือนมีตารางของพิกเซลอยู่ข้างใน นำตัวอักษรที่ตัดแต่ละตัวนั้นมาติดลงบนกระดาษกราฟนั้นและทำการล้อมกรอบสี่เหลี่ยมรอบตัวอักษรนั้นดังแสดงในรูปที่ 3.1 การล้อมกรอบสี่เหลี่ยมรอบตัวอักษรนี้ก็จะเหมือนการสร้างกล่อง สำหรับบรรจุตัวอักษร จากนั้นก็จะได้รูปแบบที่ใช้ในการสร้างตัวอักษรได้ต่อไป

รูปที่ 3.1 แสดงการใช้กระดาษกราฟเป็นแกนอ้างอิงในการสร้างตัวอักษร

หลังจากที่ได้รูปแบบตัวอักษรที่ต้องการแล้ว พบว่าตัวอักษรไทยนั้นหลายๆตัวจะมีลักษณะบางอย่างที่คล้ายคลึงกันเช่น ก ถ ภ ฤ ฦ ญ ณ ฌ เป็นต้น จะพบว่าโครงสร้างส่วนที่เป็นส่วนโค้ง หรือหลังคาของตัวอักษรเหล่านี้จะมีความคล้ายกันมากซึ่งอาจจะเรียกใช้ชุดคำสั่งแบบเดียวกัน นอกจากนี้ในส่วนหัวของตัวอักษรไทยก็จะมีรูปแบบที่เป็นอย่างเดียวกันด้วย การสร้างชุดคำสั่งที่ใช้เขียนส่วนหัวของตัวอักษรตัวหนึ่งจะสามารถนำไปใช้กับตัวอักษรตัวอื่นได้ด้วย อย่างไรก็ตามเนื่องจากความกว้างของตัวอักษรแต่ละตัวมีขนาดไม่เท่ากัน ถึงแม้ว่าจะใช้ตำแหน่งอ้างอิงเดียวกัน แต่จำนวนพิกเซลที่อยู่ระหว่างระยะห่างของจุดต่างๆจะไม่เท่ากัน ดังนั้นภาพของตัวอักษรที่ปรากฏออกมาอาจจะมีความแตกต่างกันอยู่บ้าง

เมื่อจะเริ่มต้นสร้างตัวอักษรภาษาไทยนั้น จะต้องมีการกำหนดรหัสแอสกีให้กับตัวอักษรแต่ละตัวตลอดจนมิติ ความกว้าง ยาว ลึก ของตัวอักษรตัวนั้นๆ เสียก่อน โปรแกรมเมตาฟอนต์จะมีคำสั่ง "beginchar (xxx,width,height,depth)" โดย 'xxx' คือรหัสอักขระภาษาไทยตามที่ใช้กันอยู่ทั่วไป สำหรับ width, height, depth ก็คือมิติของกล่องที่กล่าวถึงในบทที่ 2 การจบชุดคำสั่งสำหรับตัวอักษรแต่ละตัวนั้นเมตาฟอนต์ จะมีคำสั่ง "endchar" เพื่อเป็นตัวบอกการสิ้นสุดกล่องสำหรับตัวอักษรแต่ละตัว

การสร้างตัวอักษรตามรูปแบบนั้นจะสามารถทำได้โดยกำหนดจุดต่างๆที่แนวเส้นของ ตัวอักษรจะต้องผ่านและทำการเลือกขนาดและมุมของปากกาที่จะใช้เขียนตัวอักษรนั้น จากนั้นก็จะใช้คำสั่งเขียนเส้นซึ่งอาจจะเป็นเส้นตรงหรือเส้นโค้งตามรูปแบบของตัวอักษรนั้นๆ ชุดคำสั่งทั้งหมดที่สร้างขึ้นนี้จะถูกเก็บอยู่ในแฟ้มข้อมูลที่มีนามสกุลเป็น .mf

วิธีการสร้างตัวอักษร “ก” ดังที่แสดงในรูปที่ 3.1 นั้นสามารถอธิบายได้คือ ในขั้นแรกจะทำการแบ่งองค์ประกอบของตัวอักษร “ก” ออกเป็น 4 ส่วนคือ ลำตัวด้านซ้าย ขีดขวางส่วนปาก โค้งหลังคา และลำตัวด้านขวา จากนั้นก็จะทำการสร้างองค์ประกอบทีละส่วน จากการพิจารณาองค์ประกอบในส่วนแรกหรือลำตัวด้านซ้ายนั้นจะพบว่าคือสี่เหลี่ยมผืนผ้าที่มีการระบายสีดำลงไปในรูปสี่เหลี่ยมนั้น ดังนั้นสิ่งที่จะต้องกระทำคือการกำหนดพิกัดของสี่เหลี่ยมผืนผ้าดังกล่าวลงในกล่องที่มีความกว้าง 3.75 pt# ยาว 5 pt# สูง 0 pt# ซึ่งในที่นี้กำหนดให้เป็น z1l,z1r,z2l,z2r ตามลำดับ ขั้นตอนต่อไปก็จะทำการลากเส้นเชื่อมพิกัดดังกล่าวให้เกิดเป็นรูปสี่เหลี่ยมผืนผ้าและระบายสีดำลงไป เมื่อได้ลำตัวด้านซ้ายแล้วก็จะทำการสร้างองค์ประกอบที่เหลือซึ่งจะพบว่าการสร้างขีดขวางส่วนปากก็จะมีวิธีการเช่นเดียวกันกับการสร้างลำตัวด้านซ้ายดังได้กล่าวมาแล้ว

สำหรับโค้งหลังคานั้นจะมีความแตกต่างออกไปบ้างกล่าวคือเราจะทำการกำหนดตำแหน่ง 6 ตำแหน่ง ซึ่งเป็นองค์ประกอบของส่วนโค้งคือ z5l,z5r,z6l,z6r,z7l,z7r โดยที่ตำแหน่งของ z5,z6,z7 จะวางตัวกันเป็นส่วนโค้งของวงกลม เมื่อกำหนดตำแหน่งได้แล้วก็จะทำการลากกรอบของเส้นโค้งและทำการระบายสีลงไปเช่นเดิม องค์ประกอบของลำตัวด้านขวานั้นก็เช่นเดียวกับลำตัวด้านซ้ายแต่จะต่างกันเล็กน้อยตรงปลายลำตัวด้านล่างจะมีการตัดให้เฉียงลงอันเป็นลักษณะเฉพาะตัวของฟอนต์ชนิดนี้ ดังนั้นจะต้องทำการกำหนดตำแหน่งพิกัดสำหรับกรอบให้มี 6 ตำแหน่งจากนั้นก็จะเข้าสู่ขั้นตอนเดิมคือการลากเส้นกรอบและระบายสีลงในกรอบนั้นๆ

ขั้นตอนของการสร้างตัวอักษร “ก” ดังกล่าวสามารถแสดงได้ด้วยอัลกอริธึมดังนี้

```
beginchar (161,3.75pt#,5pt#,0); %Gor-Gai%
pickup pencircle scaled .6pt;

%ลำตัวด้านซ้าย
z1=(good.x .2w,good.y .65h);
z1l=lft z1;
z1r=rt z1;
z2=(good.x .2w,good.y 0h);
z2l=lft z2;
z2r=rt z2;
rec_stem(1,2);

%ขีดขวางส่วนปาก
z3r=(good.x .05w,good.y y1);
z3l=(good.x .05w,good.y y1+.075h);
z4r=(good.x .5w,good.y y3r);
z4l=(good.x .5w,good.y y3l);
rec_stem(3,4);
```

```
%โค้งหลังคา
z5l=z3l;
z5r=(good.x x5l+.075w,good.y y5l);
z6l=(good.x .5w,good.y h);
z6r=(x6l,good.y h-.05h);
z7l=(good.x .95w,y3l);
z7r=(good.x .95w-.15w,y7l);
gor_curve(5,6,7);

%ลำตัวด้านขวา
z8l=z7r;
z8r=z7l;
z9l=(x8l,good.y .1h);
z9r=(x8r,y9l);
z10l=(x9l,good.y (y9l-.1h));
z10r=(1/3[x9l,x9r],good.y y10l);
rec_sharp_stem(8,9,10);

labels (1,2);
labels (3,4);
labels (5,6,7);
labels (8,9,10);

endchar;
```

สำหรับตัวอักษรตัวอื่นนั้นก็จะใช้วิธีการสร้างแบบเดียวกันกล่าวคือจะต้องมีการแยกองค์ประกอบต่างๆ ของตัวอักษรแต่ละตัวก่อนจากนั้นก็จะทำการสร้างทีละส่วนเช่นเดียวกันกับวิธีการสร้างตัวอักษร "ก" ดังที่อธิบายไว้ข้างต้น ความลำบากของการสร้างตัวอักษรแต่ลำตัวก็คือการกำหนดพิกัดให้เหมาะสมเพื่อที่จะให้ตัวอักษรที่ออกแบบมีความสวยงามตรงตามความต้องการของผู้ออกแบบไว้นั่นเอง อย่างไรก็ตามตัว

โปรแกรมเมตาฟอนต์จะมีชุดคำสั่งสำหรับสร้างคำสั่งมาโครไว้ให้ผู้ออกแบบสามารถทำงานได้สะดวกรวดเร็วยิ่งขึ้น เช่นจากตัวอย่างข้างต้นจะเห็นมีชุดคำสั่ง rec_stem(1,2), gor_curve(5,6,7), rec_sharp_stem(8,9,10) ซึ่งเป็นคำสั่งมาโครที่เขียนขึ้นเองสำหรับใช้ในการเขียนองค์ประกอบแต่ละส่วน ดังนั้นถ้าต้องการสร้างตัวอักษร "ถ" ก็ไม่จำเป็นที่จะต้องเขียนชุดคำสั่งเหล่านี้ซ้ำอีก เพียงแต่กำหนดพิกัดใหม่ขึ้นและสร้างชุดคำสั่งสำหรับองค์ประกอบที่ตัวอักษร "ก" ยังไม่มี ในที่นี้คือส่วนหัวขึ้นใหม่ จากนั้นทำการเรียกใช้ชุดคำสั่งมาโครที่มีอยู่แล้วก็จะทำให้สามารถประหยัดเวลาในการสร้างตัวอักษรลงได้เป็นอย่างมาก

โดยปกติแล้วลายเส้นของตัวอักษรส่วนใหญ่ในฟอนต์ภาษาไทยจะถูกกำหนดให้อยู่ในกล่อง ซึ่งจะใช้ค่าพิกัดอ้างอิงเป็นบวก แต่จะมีกลุ่มของสระหรือวรรณยุกต์บางตัวที่อยู่เหนือหรือใต้เส้นบรรทัด ปกติ กลุ่มของตัวอักษรเหล่านี้จะพิกัดอ้างอิงสำหรับมิติความกว้างเป็นลบ และความกว้างของกล่องมีค่าเป็นศูนย์ที่เป็นเช่นนี้เนื่องจากการทำงานของเท็กซ์จะเป็นการนำกล่องของตัวอักษรใช้งานตัวแต่ละตัวมาเรียงต่อกันทีละกล่อง เมื่อมิติของตัวอักษรนั้นเป็นลบ กล่องที่นำมาเรียงจึงเป็นกล่องว่าง แต่ตัวอักษรที่อ้างอิงกับกล่องนั้นจะอยู่ถัดไปด้านซ้ายมือของกล่องว่าง ซึ่งจะไปวางอยู่บนหรือใต้กล่องใบที่วางไว้ก่อนแล้วจึงเสมือนกับการจัดบรรทัดสำหรับตัวอักษรไทยทั้ง 3 ระดับไปในตัว อย่างไรก็ตามถ้ากล่องว่างนั้นยังมีความกว้างอยู่บ้าง ถ้าตัวอักษรถัดไปเป็นวรรณยุกต์อีกจะทำให้วรรณยุกต์ตัวนั้นถูกวางอยู่บนกล่องว่างที่วางไว้อยู่ก่อนหน้านี้ เมื่อผ่านการดำเนินงานของเท็กซ์จะปรากฏเป็นวรรณยุกต์ตัวนั้นถูกวางอยู่ลอยๆ โดยไม่มีพยัญชนะหรือสระรองรับอยู่ด้านล่างซึ่งผิดรูปแบบภาษาไทยวิธีการแก้ปัญหานี้จึงต้องทำให้ความกว้างของกล่องสำหรับวรรณยุกต์และสระที่อยู่เหนือหรือใต้เส้นบรรทัดมีค่าเป็นศูนย์

เมื่อสร้างภาพตัวอักษรนั้นเสร็จแล้ว ก็สามารถตรวจสอบหน้าตาของตัวอักษรนั้นได้ โดยใช้โปรแกรมเมตาฟอนต์เรียกดังนี้

สมมุติว่าแฟ้มนั้นชื่อ thai.mf เราจะทำการเรียกโดยใช้คำสั่ง

```
C:\>mf thai
```

โปรแกรมเมตาฟอนต์จะทำการสร้างภาพตัวอักษรทั้งหมดที่ถูกสร้างขึ้นบนหน้าจอ ในขณะเดียวกันก็จะทำการเก็บข้อมูลของตัวอักษรเหล่านั้นไว้ในแฟ้มตัวอักษรแบบทั่วไป (generic font file) ที่มีนามสกุลเป็น

.2602gf เพื่อเปลี่ยนเป็นแฟ้มที่ไดรว์เวอร์รู้จักและแฟ้มเท็กซ์ฟอนต์ เมตริกที่มีนามสกุลเป็น .tfm สำหรับเท็กซ์เรียกใช้งานต่อไป

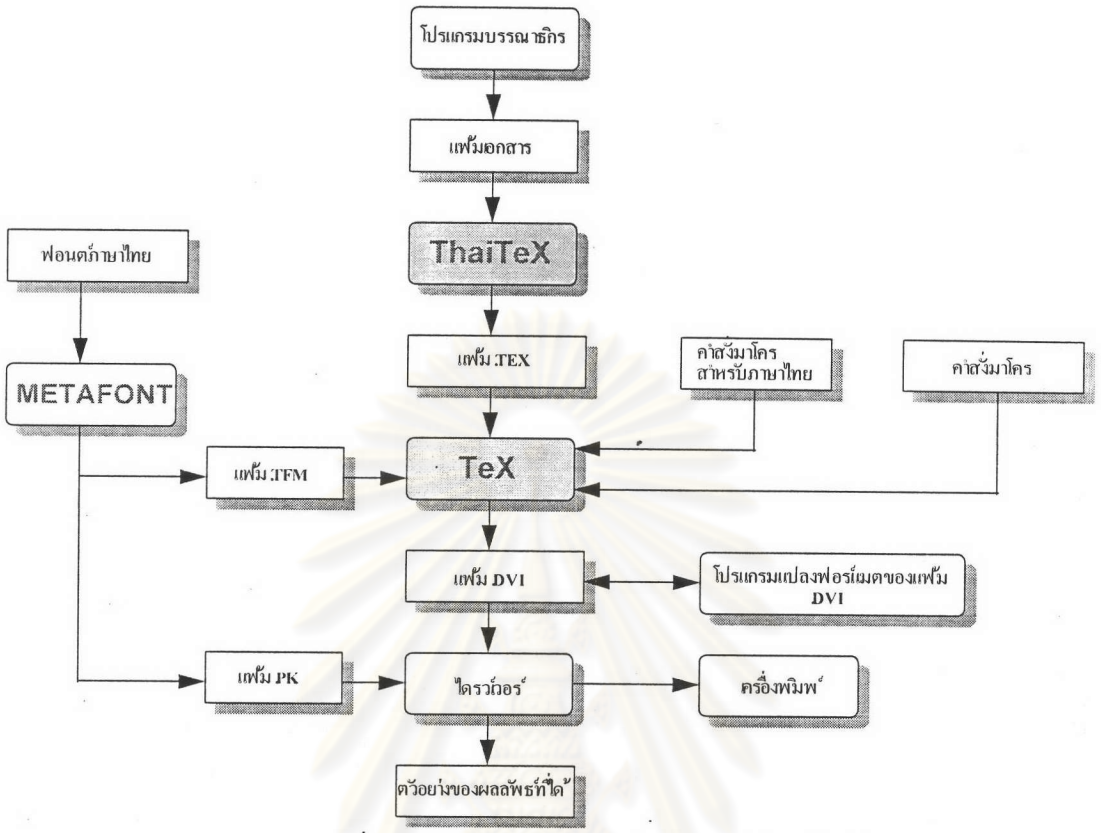
เมื่อได้เห็นภาพตัวอักษรเหล่านี้แล้วก็สามารถที่จะแก้ไขหรือตกแต่งความสวยงามเพิ่มเติมได้ตามต้องการ หลังจากที่แก้ไขรายละเอียดของโปรแกรมเรียบร้อยแล้วก็จะทำการสร้างแฟ้มตัวอักษรแบบทั่วไปขึ้นมาใหม่โดยเปลี่ยนโหมดการทำงานของเมตาฟอนต์เป็นโหมดโลคอลฟอนต์ (localfont mode) ซึ่งจะเป็นโหมดที่มีการกำหนดค่าเรสโซรูชันและจำนวนพิกเซล สำหรับอุปกรณ์ ที่จะใช้เป็นเอาท์พุตจริง และอาจจะกำหนดอัตราขยายขนาดหรือย่อขนาด (magnification) ตามความพอใจได้ ในครั้งนี้แฟ้มตัวอักษรแบบทั่วไปจะมีนามสกุลเป็น .300gf แต่แฟ้มเท็กซ์ฟอนต์เมตริก (.tfm) นั้นจะยังเหมือนเดิม หลังจากนั้นจะทำการเปลี่ยนแฟ้ม .300gf ให้เป็นแฟ้ม .pk สำหรับ ที่จะเรียกใช้งานในขั้นตอนต่อไป

2. <u>การประมวลผลแฟ้มเอกสารภาษาไทยสำหรับโปรแกรมเท็กซ์</u>

สิ่งที่เท็กซ์จะรู้จักในแฟ้มเอกสารที่จะทำการประมวลผลก็คือรหัสแอสกี้ ดังที่ทราบมาแล้วว่าในแฟ้มเอกสารหนึ่งๆที่จะทำการประมวลผลนั้นจะประกอบไปด้วยชุดคำสั่งมาโครและตัวข้อมูลปะปนกันไปชุดคำสั่งจะเป็นตัวสั่งให้เท็กซ์ทำงานต่างๆ ในขณะที่ตัวข้อมูลก็คือเนื้อหาที่เราต้องการพิมพ์ คำถามที่เกิดขึ้นก็คือว่าเท็กซ์จะรู้ได้อย่างไรว่าอะไรเป็นข้อมูลและ อะไรเป็นชุดคำสั่ง

โดยเนื้อแท้ของเท็กซ์แล้วก็จะทำงานเหมือนกับตัวแปลภาษาชั้นสูงหรือคอมไพเลอร์ โดยจะแยกความแตกต่างระหว่างสิ่งที่เป็นคำสั่งและสิ่งที่เป็นข้อมูลด้วยเครื่องหมายอย่างใดอย่างหนึ่ง ในกรณีของเท็กซ์นี้จะใช้เครื่องหมายแบ็คแสลช (backslash) " ในการบอกจุดเริ่มต้นของคำสั่งเมื่อได้รับคำสั่งอย่างใดอย่างหนึ่งภายใต้เครื่องหมายแบ็คแสลชแล้ว เท็กซ์ก็จะไปดำเนินการตามคำสั่งนั้น สำหรับสิ่งที่ไม่ได้นำหน้าด้วยแบ็คแสลชนั้นเท็กซ์จะถือว่าเป็นข้อมูลหรือตัวอักษร วิธีที่จะจัดการกับตัวอักษรเหล่านี้ก็คือเท็กซ์จะทำการตรวจสอบว่าในขณะนี้มีการเรียกใช้ฟอนต์ชุดใดก็จะไปนำตัวอักษรในฟอนต์ชุดนั้นมาวางไว้และจัดรูปแบบตามคำสั่งของมาโครที่กำหนด ทำเช่นนี้ไปเรื่อยๆจนจบแฟ้มเอกสาร ถ้าฟอนต์ชุดนั้นเป็นตัวอักษรภาษาไทย เท็กซ์ก็จะนำตัวอักษรไทยมาวางไว้เช่นเดียวกัน

ขั้นตอนการทำงานของโปรแกรมไทยเท็กซ์สามารถแสดงได้ดังรูปที่ 3.2



รูปที่ 3.2 แสดงการทำงานของโปรแกรมไทยเท็กซ์

2.1 *รับอินพุตจากแฟ้มเอกสาร*

แฟ้มเอกสารที่จะใช้เป็นอินพุตในขั้นตอนนี้จะเหมือนกันกับแฟ้มเอกสารที่เท็กซ์ใช้ โดยทั่วไปในขั้นตอนนี้ก็เพียงแต่ใช้โปรแกรมบรรณาธิกรทั่วไป โดยทำการรันโปรแกรมไทยไดรว์เวอร์ก่อนการรันโปรแกรมบรรณาธิกรนั้น โปรแกรมบรรณาธิกรที่ใช้จะต้องไม่มีส่วนของหัวแฟ้ม (file header) สลับซับซ้อน เนื่องจากเมื่อผ่านขบวนการของการแปลงให้อยู่ในรูปแบบของเท็กซ์แล้วจะเกิด สัญญลักษณ์ที่เท็กซ์ไม่รู้จัก

2.2 *ทำการแทรกเครื่องหมายของการตัดคำไว้สำหรับแต่ละบรรทัด*

อัลกอริธึมของ CU Writer จะทำการอ่านแฟ้มข้อมูลเข้ามาทีละบรรทัด เพื่อทำการหาจุดแบ่งคำ จากนั้นจะดำเนินการแทรกเครื่องหมายตัดคำ (delimiter) สำหรับแต่ละบรรทัด

2.3 *ปรับเปลี่ยนตัวอักษรเพื่อความสวยงาม*

ผลลัพธ์ที่ได้จากอัลกอริธึมตัดคำ จะถูกเก็บอยู่ในอะเรย์ของสตริงโดยจะเก็บเรียงติดต่อกันไปทั้งพยัญชนะสระ วรรณยุกต์ แทรกด้วยเครื่องหมายแบ่งคำเป็นระยะๆไป ซึ่งผลลัพธ์ที่ได้นี้ก็เพียงพอที่จะนำไปเป็นอินพุตสำหรับเท็กซ์ได้แล้ว (ถ้ามีการแทรกชุดคำสั่งมาโครในตำแหน่งที่เหมาะสม) แต่ผลลัพธ์ที่ออกมาจะไม่สวยงามเนื่องจากระยะห่างระหว่างตัวพยัญชนะนั้นบางครั้งจะเกิดการทับซ้อนกันได้ ถ้าต้องการความสวยงามจะต้องดำเนินการแก้ไขในส่วนนี้ด้วย

ในโปรแกรมเท็กซ์จะมีวิธีการปรับเปลี่ยนตัวอักษรภาษาอังกฤษที่ใช้เพื่อความสวยงามเรียกว่าเคิร์นนิงและลิเกเชอร์ (kerning and ligature) ในการออกแบบไทยเท็กซ์ก็ จะมีหลักการนี้เข้ามาเกี่ยวข้องด้วยเช่นกัน

เคิร์นนิง (kerning) คือ การปรับเปลี่ยนขนาดช่องว่างระหว่างตัวอักษรที่ใช้ให้แคบลงหรือกว้างขึ้นเพื่อความสวยงาม ตามปกติแล้วการพิมพ์ตัวอักษรแต่ละตัวจะมีช่องว่างระหว่างตัวอักษรที่ตายตัวเรียกว่าช่องไฟ แต่เนื่องจากตัวอักษรแต่ละตัวมีขนาดของกล่องไม่เท่ากัน การเว้นช่องไฟที่เท่ากันจะทำให้การพิมพ์คู่ของตัวอักษรบางตัวดูไม่สวยงามเนื่องจากมีช่องว่างระหว่างตัวอักษรมากหรือน้อยเกินไป เช่นระหว่างตัวอักษร "W" และ "e" จะมีช่องว่างมากกว่าตัวอักษร "M" และ "e" เนื่องจาก องค์ประกอบด้านขวาสุดของตัว "W" จะเอียงขึ้นทางขวาทำให้มีพื้นที่ว่างใต้ตัวอักษรนั้นมากเกินไป เท็กซ์จึงแก้ปัญหานี้โดยการทำเคิร์นนิงให้ตัว "e" เลื่อนชิดเข้ามาอยู่ภายใต้เงาของ "W" ดังรูปที่ 3.3

We We

รูปที่ 3.3 แสดงความแตกต่างระหว่างก่อนทำเคิร์นนิง(ด้านซ้ายมือ)กับหลังทำเคิร์นนิง

ในภาษาไทยนั้น เนื่องจากในภาษาไทยจะมีตัวอักษรบางตัวที่มีส่วนหางยาวเกินกว่าความสูงของตัวอักษรปกติ เช่น "ป" "ฟ" "ฝ" ตัวอักษรที่มีหางยาวต่ำกว่าบรรทัดปกติ เช่น "ฎ" "ฏ" หรือสระที่มีความสูงมากกว่าสระอื่นเช่น "า" เมื่อนำมาใช้กับสระหรือวรรณยุกต์ที่อยู่สูงหรือต่ำกว่าแนวตัวอักษรปกติที่ใช้แล้วจะทำให้เกิดการทับซ้อนกันของหางตัวอักษรหรือสระนั้น กับสระหรือวรรณยุกต์ที่อยู่สูงหรือต่ำกว่า จึงจำเป็นที่

จะต้องแก้ไขโดยการเปลี่ยนรหัสของสระและวรรณยุกต์ที่ตามหลังตัวอักษรดังกล่าวไปใช้รหัสในตำแหน่งที่จะไม่ทำให้เกิดการทับซ้อน อย่างไรก็ตามวิธีการเช่นนี้จะไม่เหมือนกับการทำเคิร์นนิงของเท็กซ์ในภาษาอังกฤษเพราะในภาษาอังกฤษนั้นการทำเคิร์นนิงจะไม่ทำให้รหัสของตัวอักษรที่ใช้เปลี่ยนไปแต่จะใช้ความสามารถของโปรแกรมในการลดหรือขยายขนาดของช่องไฟแทน ดังนั้นอาจจะเรียกวิธีการที่เราใช้สำหรับเลื่อนตำแหน่งของสระหรือวรรณยุกต์ของภาษาไทยนี้ว่าเป็นการทำเคิร์นนิงปลอมก็ได้

ลิเกเชอร์ (ligature) คือ การแทนที่กลุ่มของตัวอักษรบางตัวที่มีบางส่วน คล้ายคลึงหรือกลมกลืนกันด้วยตัวอักษรพิเศษตัวหนึ่งแล้วทำให้ตัวอักษรกลุ่มนั้นดูสวยงามขึ้นเช่น ระหว่าง ตัวอักษร "f" และ "i" จะมีความกลมกลืนกันระหว่างองค์ประกอบส่วนปลายของ "f" กับ จุดเหนือ "i" ดังนั้นจึงต้องทำลิเกเชอร์ระหว่างตัวอักษรคู่นี้ โดยใช้ตัวอักษรชุดใหม่แทนที่ ดังรูปที่ 3.4

fi fi

รูปที่ 3.4 แสดงภาพก่อนทำลิเกเชอร์(ด้านซ้ายมือ)และหลังทำลิเกเชอร์

ในภาษาไทยนั้น ถึงแม้ว่าจะไม่มีกลุ่มตัวอักษรที่ต้องทำลิเกเชอร์เช่นเดียวกับตัวอย่างที่กล่าวข้างต้น แต่อย่างไรก็ตามจะมีตัวอักษรบางตัวที่เมื่อนำมาประกอบกับสระแล้วจะต้องมีการลดรูปตัวอักษรนั้นลง ตัวอักษรดังกล่าวได้แก่ "ฐ" และ "ญ" เมื่อ นำมาประกอบกับสระที่อยู่ต่ำกว่าบรรทัดปกติ ส่วนของฐานจะหายไป ซึ่งในการออกแบบการทำงานของโปรแกรมไทยเท็กซ์จะต้องทำการตรวจสอบในส่วนนี้ด้วย โดยเมื่อโปรแกรมตรวจพบตัวอักษรดังกล่าวแล้วจะต้องทำการตรวจสอบตัวอักษรถัดไปว่าเป็นสระที่อยู่ใต้บรรทัดหรือไม่ถ้าใช่ก็จะทำการเปลี่ยนรหัสของตัวอักษรปกติให้เป็นรหัสของตัวอักษรที่มีการลดรูปแล้ว วิธีการเช่นนี้ก็อาจจะเรียกว่าเป็นลิเกเชอร์ปลอมได้เช่นกัน

การทำเคิร์นนิงและลิเกเชอร์ในภาษาอังกฤษนั้นจะทำในแนวระดับ แต่สำหรับภาษาไทยแล้วจะกระทำในแนวดิ่ง เนื่องจากภาษาไทยเป็นภาษาที่มีตัวอักขระอยู่ใน 3 ระดับ วิธีการทำจะต้องนำเอาอะเรย์ของ

สตริงที่ได้จากอัลกอริธึมตัดคำมาแบ่งออกเป็น 4 ระดับ คือระดับล่าง ระดับกลาง ระดับสูง และระดับสูงสุด โดยอะเรย์เหล่านั้นจะทำหน้าที่เก็บ สระที่อยู่ต่ำกว่าบรรทัด พยัญชนะและที่สระที่อยู่ในบรรทัดปกติ สระที่อยู่เหนือระดับบรรทัด และวรรณยุกต์ ตามลำดับ

อัลกอริธึมสำหรับการทำเคิร์นนิงและลิเกเชอร์สำหรับภาษาไทยแสดงได้ดังนี้

```
/*ตารางบอกระดับของสระและวรรณยุกต์ ตั้งแต่ 0xD0-0xFF*/
/*0 คือ ระดับปกติ*/
/*1 คือ ระดับล่าง*/
/*2 คือ ระดับสูง*/
/*3 คือ ระดับสูงสุด*/
int levtable[] =
        0,2,0,0,2,2,2,2,1,1,1,0,0,0,0,0,
        0,0,0,0,0,0,0,2,3,3,3,3,3,2,3,0,
        0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0 ;
/*ตารางบอกการเปลี่ยนตัวสระและวรรณยุกต์*/
int lefttab[] = 136,131,    /*คือการเปลี่ยนรหัสจาก 136 เป็น 131*/
        137,132,
        138,133,
        139,134,
        140,135,
        0xED,0x8F,
        0xE8,0x98,
        0xE9,0x99,
        0xEA,0x9A,
        0xEB,0x9B,
        0xEC,0x9C,
        0xD4,0x94,
        0xD5,0x95,
```

```
            0xD6,0x96,
            0xD7,0x97,
            0xD1,0x92,
            0xE7,0x93;

/*ส่วนแรกนำค่าที่อยู่ในอะเรย์ที่ได้จากการตัดคำมาแยกใส่อะเรย์แบ่งระดับ*/
for(i=0;i<4096;i++)
  top[i]=up[i]=middle[i]=low[i]=0;
i=0; k=-1;
while((c=line[i++])!=0)
  switch((c>0xD0)?levtable[c-0xD0]:0)
  case 0: /*กรณีที่อยู่ในบรรทัดปกติ*/
     if(c==0xD3)  /*กรณีสำหรับสระ "า"*/
      if(k>=0)
       up[k]=0xED;

      k++;
      middle[k]=0xD2;

     else  /*กรณีสระอื่นๆที่อยู่ในบรรทัดปกติ*/
      k++;
      middle[k]=c;

      break;
  case 1: /*กรณีสระที่อยู่ใต้บรรทัดปกติ*/
     low[k]=c; break;
  case 2: /*กรณีสระที่อยู่เหนือบรรทัดปกติ*/
     up[k]=c; break;
  case 3: /*กรณีวรรณยุกต์*/
```

```
        top[k]=c; break;


/*ส่วนที่สองการปรับแต่งตัวอักษรเพื่อความสวยงาม*/
for(i=0;i<=k;i++)
 /*เคิร์นนิงปลอมปรับแต่งวรรณยุกต์กรณีที่ไม่มีสระอยู่เหนือบรรทัดปกติ*/
  if((top[i]&&(up[i]==0))
    up[i] = top[i]-96;
    top[i] = 0;

 /*เคิร์นนิงปลอมของ "ป","ฝ","ฟ"*/
  if(middle[i] == 0xBB ||
     middle[i] == 0xBD ||
     middle[i] == 0xBF)
     if(up[i]) up[i] = moveleft(up[i]);
     if(top[i]) top[i] = moveleft(top[i]);

 /*เคิร์นนิงปลอมของ "ฎ","ฏ"*/
  if(middle[i] == 0xAE || middle[i] == 0xAF)
    if(low[i]) low[i] = low[i] + 36;

 /*ลิเกชันปลอมของ "ฐ","ญ"*/
  if(middle[i] == 0xB0 && low[i])
    middle[i] = 0x9F;
  if(middle[i] == 0xAD && low[i])
    middle[i] = 0x90;

/*ส่วนที่สาม การนำอะเรย์แบ่งระดับใส่กลับคืนเข้าไปในอะเรย์ตัดคำเดิม*/
i=0;k=0;
```

```
while(middle[i])
 line[k++]=middle[i];
 if(low[i]) line[k++]=low[i];
 if(up[i])  line[k++]=up[i];
 if(top[i]  line[k++]=top[i];
 i++;

line[k]=0;
```

การปรับเปลี่ยนตัวอักษรเพื่อความสวยงามนี้แบ่งเป็นสามส่วน ในส่วนแรกจะนำค่าจากอะเรย์ที่ได้จากการตัดคำมาบรรจุไว้ในอะเรย์แบ่งระดับ 4 อะเรย์ ในส่วนที่สองจะเริ่มทำการตรวจสอบเคิร์นนิงและลิเกชัน โดยถ้าพบว่าเป็นตัวอักษรที่สมควรทำก็จะทำการเปลี่ยนตัวอักษรนั้นไปใช้ตัวอักษรในตำแหน่งที่เหมาะสมกว่าแทน ในส่วนที่สามจะเป็นการนำอะเรย์ทั้งสี่เก็บคืนเข้าไปยังอะเรย์ตัดคำเดิมเพื่อนำไปใช้สำหรับตรวจสอบและแทรกคำสั่งมาโครสำหรับการใช้ภาษาไทยต่อไป

2.4 *การแทรกคำสั่งมาโครสำหรับเรียกใช้ฟอนต์ภาษาไทย*

จากอะเรย์ตัดคำที่ผ่านการปรับเปลี่ยนตัวอักษรเพื่อความสวยงามแล้ว จะเป็นขั้นตอนการแทรกคำสั่งมาโครสำหรับเรียกใช้ฟอนต์ภาษาไทยตามที่ต้องการ

ในการใช้งานเท็กซ์โดยทั่วไปในระบบภาษาอังกฤษถ้าไม่มีการใช้คำสั่งมาโคร เพื่อเรียกใช้ฟอนต์ชนิดใดชนิดหนึ่งเป็นกรณีพิเศษแล้ว เท็กซ์จะถือเอาฟอนต์ชุดคอมพิวเตอร์โมเดิร์นฟอนต์ เป็นมาตรฐาน คอมพิวเตอร์โมเดิร์นฟอนต์ที่เท็กซ์เรียกใช้นี้จะถูกจัดเก็บอยู่ในแฟ้มที่มีชื่อว่า "CMRxx.TFM" ซึ่งพัฒนามาจากโปรแกรมเมตาฟอนต์นั่นเอง เท็กซ์จะตรวจสอบข้อมูลหรือสัญญลักษณ์ที่ต้องการจะพิมพ์ก่อนแล้วจึงนำตัวอักษรนั้นมาจากคอมพิวเตอร์โมเดิร์นฟอนต์ ดังนั้นถ้าในแฟ้มเอกสารปรากฏตัวอักษรหรือสัญญลักษณ์แปลกๆที่ไม่มีในคอมพิวเตอร์โมเดิร์นฟอนต์ เท็กซ์ก็จะผ่านไม่นำตัวอักษรดังกล่าวมาพิมพ์ให้ ในกรณีของภาษาไทยนี้ก็เช่นเดียวกัน ถ้าไม่มีการแทรกคำสั่งมาโครไว้ก่อนการใช้ภาษาไทย เท็กซ์ก็จะผ่านตัวอักษรนั้นไปโดยพิมพ์แต่ตัวอักษรที่เป็นภาษาอังกฤษหรือตัวเลขเท่านั้น ด้วยเหตุนี้โปรแกรมไทยเท็กซ์จะต้องมีส่วนของการแทรกคำสั่งมาโครสำหรับการเรียกใช้ฟอนต์ภาษาไทยอยู่ด้วย

การแทรกคำสั่งสำหรับเรียกใช้ฟอนต์จะใช้คำสั่ง "{\thai" วางไว้ก่อนหน้าตัวอักษรไทยตัวแรกของกลุ่มคำภาษาไทยนั้นเช่นตัวอย่างประโยค

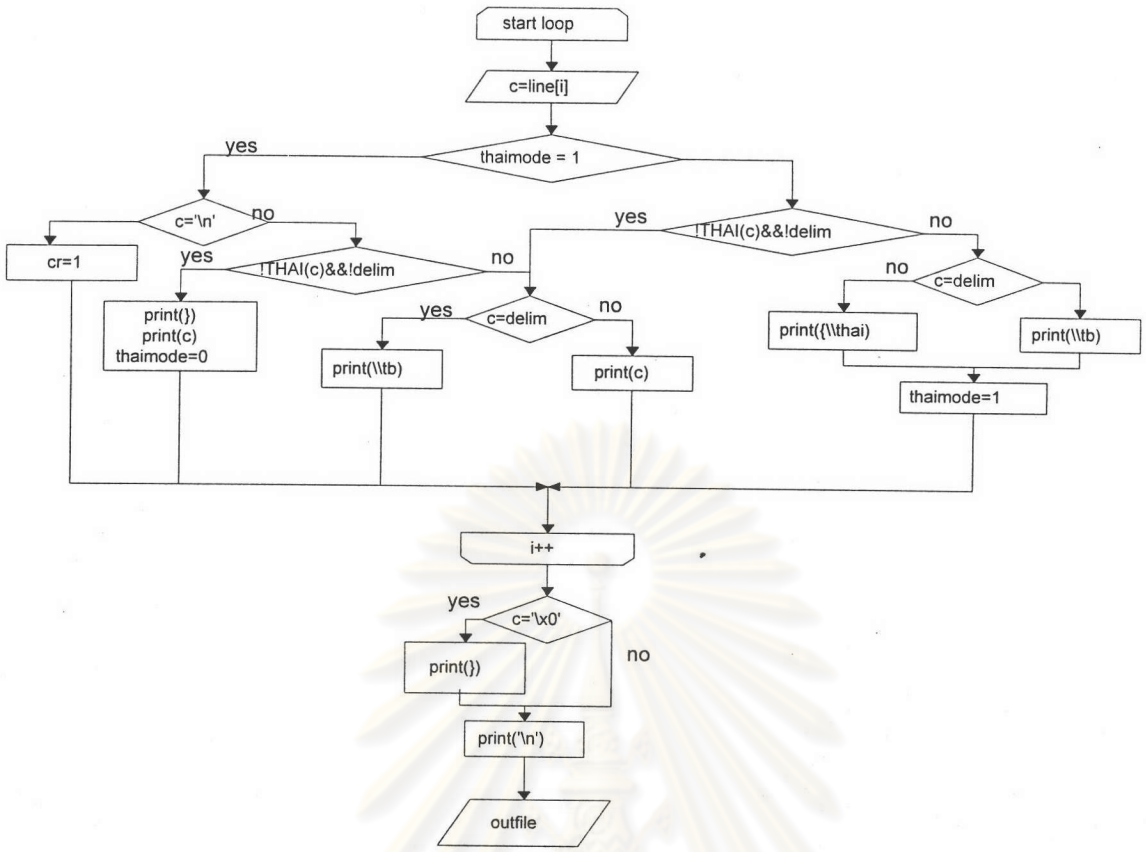
"การตัดคำภาษาไทย (Thai words separation) นี้เป็นส่วน..."

จะต้องมีการวางคำสั่งมาโครดังนี้

{\thai การ\tb ตัด\tb คำ\tb ภาษา\tb ไทย \tb} (Thai words separation) {\thai นี้\tb เป็น\tb ส่วน\tb ...

ตำแหน่งที่เหมาะที่สุดที่จะทำการแทรกคำสั่งมาโครสำหรับเรียกใช้ฟอนต์นั้นคือตำแหน่งก่อนเริ่มต้นของกลุ่มคำภาษาไทยที่อยู่ในอะเรย์และตำแหน่งหลังตำแหน่งของตัวอักษรไทยตัวสุดท้ายก่อนที่จะเปลี่ยนไปใช้ภาษาอังกฤษโดยที่จะใส่เครื่องหมาย “{\thai” ไว้หน้าตัวอักษรไทยและเครื่องหมาย “}” ไว้หลังตำแหน่งสิ้นสุดกลุ่มคำภาษาไทยในแต่ละช่วง ทั้งนี้เนื่องจากเป็นจุดสิ้นสุดใจความสำคัญในภาษาไทยพอดี นอกจากนี้การที่จะป้องกันการขึ้นบรรทัดใหม่ก่อนการสิ้นสุดของคำภาษาไทยนั้นจะใช้คำสั่ง “\tb” ซึ่งเป็นคำสั่งเดียวกันกับคำสั่ง “\-” ที่ใช้อยู่ในเท็กซ์ การทำงานของคำสั่งนี้จะบังคับไม่ให้เท็กซ์ทำการขึ้นบรรทัดใหม่ระหว่างตำแหน่งของคำที่อยู่ระหว่าง “\-” การขึ้นบรรทัดใหม่จะกระทำได้ในตำแหน่งของ “\-” เท่านั้น การแทรกตำแหน่งของเครื่องหมาย “\tb” นี้ก็คือตำแหน่งของจุดแบ่งคำที่ได้จากอัลกอริธึมแบ่งคำนั่นเอง

ขั้นตอนการทำงานของอัลกอริธึมนี้สามารถแสดงได้ดังรูปที่ 3.5



รูปที่ 3.5 แสดงขั้นตอนการทำงานของอัลกอริธึมแทรกคำสั่งมาโคร

การทำงานจะแบ่งออกเป็นสองส่วน ส่วนแรกจะเป็นส่วนของการบอกตำแหน่งที่เริ่มล้อมกรอบตัวอักษรที่เป็นภาษาไทย ในส่วนนี้จะเริ่มจากการค้นหาตำแหน่ง 3 ตำแหน่งคือ ตำแหน่งที่สิ้นสุดคำภาษาอังกฤษ ตำแหน่งที่บอกจุดแบ่งคำ หรือตำแหน่งเริ่มต้นบรรทัดใหม่ในกรณีที่ตัวอักษรที่อ่านได้ก่อนหน้านี้เป็นภาษาไทย เมื่อพบตำแหน่งดังกล่าวข้างต้นนี้ก็จะทำการแทรกสัญญลักษณ์ "{\thai" เข้าไว้ข้างหน้าก่อนที่จะทำการพิมพ์ตัวอักษรที่อ่านมาได้นั้นลงไป สำหรับในส่วนที่สองจะเป็นส่วนของการบอกตำแหน่งสิ้นสุดของการล้อมกรอบสำหรับตัวอักษรไทย ตำแหน่งที่จะบอกการสิ้นสุดนี้ก็จะมี 3 ตำแหน่งเช่นกันคือ ตำแหน่งที่ก่อนเริ่มต้นคำภาษาอังกฤษ ตำแหน่งที่บอกจุดแบ่งคำ หรือตำแหน่งที่สิ้นสุดบรรทัดในกรณีที่ตัวอักษรก่อนหน้านี้เป็นภาษาไทย เมื่อพบตำแหน่งดังกล่าวแล้วก็จะทำการแทรกสัญญลักษณ์ "}" ลงไป ผลลัพธ์ที่ได้จากขั้นตอนนี้จะถูกเก็บไว้ในแฟ้มที่มีนามสกุลเป็น .tex เพื่อให้เท็กซ์ใช้ประมวลผลต่อไป

3. <u>การสร้างชุดคำสั่งมาโครเพื่อจัดการกับรูปแบบภาษาไทย</u>

ขั้นตอนนี้เป็นขั้นตอนของการนำความรู้เกี่ยวกับคำสั่งมาโครของเท็กซ์มาสร้างชุดคำสั่งจัดเตรียมรูปแบบของการพิมพ์ ซึ่งสามารถดัดแปลงได้หลายรูปแบบขึ้นอยู่กับลักษณะของงานที่ใช้ว่าจะเป็นในรูปแบบใด ในขั้นตอนนี้จะมีเรื่องที่เข้ามาเกี่ยวข้องมากมายเช่นการกำหนดชนิดและขนาดของฟอนต์เพื่อใช้งาน การกำหนดระยะห่างระหว่างบรรทัดที่จะจัดพิมพ์ การกำหนดขนาดของแต่ละย่อหน้า รวมไปถึงการกำหนดรูปแบบของตัวอักษรที่จะใช้เป็นหัวข้อ ชื่อบท หรือหมายเหตุตอนท้ายของหน้า ขั้นตอนนี้จะไม่มีรูปแบบที่ชัดเจนตายตัวแต่จะขึ้นอยู่กับจุดประสงค์ของการใช้งานและความสวยงามตามความต้องการของผู้ใช้เป็นหลัก

รูปแบบต่างๆที่ต้องการใช้ในแฟ้มเอกสารนั้นจะถูกกำหนดเอาไว้ในแฟ้มแยกต่างหากอีกแฟ้มหนึ่ง ซึ่งในแฟ้มนั้นจะมีเฉพาะการกำหนดคำสั่งมาโครที่ต้องการใช้ขึ้นมาใหม่เท่านั้น การเรียกใช้คำสั่งมาโครจะกระทำได้โดยพิมพ์ \input "<ชื่อแฟ้มมาโคร>" ไว้ที่ตอนต้นของแฟ้มเอกสารที่จะใช้ประมวลผลเท่านั้น