บทที่ 2

โปรแกรมอรรถประโยชน์ด้านการแปลงโครงสร้างแฟ้มข้อมูล

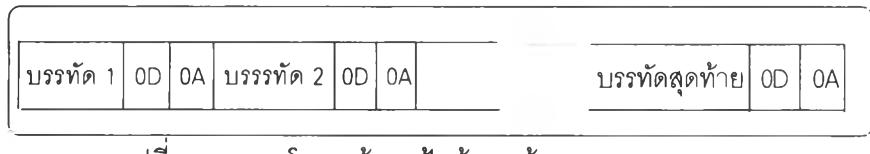
บทนี้จะกล่าวถึงแนวคิดเหตุผลและทฤษฎีที่ใช้ในการพัฒนาโปรแกรมอรรถประโยชน์สำหรับจัดการแฟ้มเอกสารของจุฬาจารึก เพื่อเสริมคุณสมบัติให้จุฬาจารึก ทางด้านการแปลงโครงสร้างแฟ้มข้อมูล นอกจากนี้ยังกล่าวถึงการออกแบบและพัฒนาโปรแกรมอรรถประโยชน์ด้านการแปลงโครงสร้างแฟ้มข้อมูล บทนี้จึงแบ่งเป็น 3 ส่วน ส่วนแรกกล่าวถึงโครงสร้างแฟ้มข้อมูลชนิดต่างๆที่ต้องการแปลงและโครงสร้างแฟ้มข้อมูลจุฬาจารึก 78 ส่วนที่สองจะกล่าวถึงขั้นตอนวิธีการแปลงเป็นโครงสร้างแฟ้มข้อมูลจุฬาจารึก 78 ส่วนที่สามจะกล่าวถึงโครงสร้างข้อมูลที่ใช้ในโปรแกรมและการพัฒนาโปรแกรมด้านการแปลงโครงสร้างแฟ้มข้อมูล

โครงสร้างแฟ้มข้อมูล (file format)

เมื่อเวิร์ดจุฬาได้ถูกปรับปรุงเพิ่มเติมคุณสมบัติเป็นจุฬาจารึกที่สามารถทำงานภายใต้ไมโครซอฟท์วินโดวส์ได้ แนวทางพัฒนาจุฬาจารึกนั้นยึดมุมมองของผู้ใช้เป็นหลัก เพื่อให้ผู้ใช้สามารถใช้งานจุฬาจารึกในลักษณะเดียวกับเวิร์ดจุฬา ตั้งชื่อเป็นจุฬาจารึก 77 โดยมีโครงสร้างแฟ้มข้อมูลเช่นเดียวกับโครงสร้างแฟ้มข้อมูลเวิร์ดจุฬา แต่มีโครงสร้างข้อมูลส่วนรูปภาพเพิ่มด้วย ต่อมาได้พัฒนาเป็นจุฬาจารึก 78 ให้ใช้งานได้อย่างมีประสิทธิภาพมากขึ้นโดยจำเป็นต้องเปลี่ยนโครงสร้างแฟ้มข้อมูล ทำให้ไม่สามารถนำแฟ้มข้อมูลเดิมมาใช้ได้ จึงต้องแปลงโครงสร้างแฟ้มข้อมูลเวิร์ดจุฬารุ่นที่ 1.52 และจุฬาจารึก 77 เป็นแฟ้มข้อมูลจุฬาจารึก 78 ก่อน เวิร์ดจุฬาได้รับการพัฒนาอย่างต่อเนื่องจนถึงรุ่นที่ 1.6 โดยเพิ่มความสามารถในการสร้างสมการคณิตศาสตร์ สามารถใช้แบบอักษรได้ 4 แบบ แยกพยางค์โดยใช้พจนานุกรม เมื่อเกิดการตัดคำจะแทรกรหัส 0xA0 แทนช่องว่าง (0x20) เป็นต้น เวิร์ดจุฬารุ่นที่ 1.6 ก็ได้รับความนิยมมากเช่นกันจึงควรแปลงโครงสร้างแฟ้มข้อมูลเป็นจุฬาจารึก 78 ด้วย นอกจากนี้แฟ้มข้อมูลข้อความ (text file) ที่สร้างได้จากโปรแกรมทั่วไปยังมีโครงสร้างแฟ้มข้อมูลที่ง่ายกว่าเวิร์ดจุฬา จึงควรแปลงโครงสร้างแฟ้มข้อมูลเป็นจุฬาจารึก 78 ด้วย

1. โครงสร้างแฟ้มข้อมูลข้อความ

แฟ้มข้อมูลข้อความจะจัดเก็บตัวอักษรตามรหัสแอสกี โดยเรียงลำดับทีละตัวอักษรไปเรื่อยๆจากบรรทัดแรกจนถึงบรรทัดสุดท้าย ในที่นี้จะกล่าวเฉพาะรหัสที่เกี่ยวข้องกับการแปลงโครงสร้างแฟ้มข้อมูลคือ รหัสขึ้นบรรทัดใหม่ของแต่ละบรรทัดมีขนาด 2 ไบต์ที่มีค่าฐานสิบหกเป็น 0x0D, 0x0A และรหัสตั้งระยะ (tab) มีขนาด 1 ไบต์ที่มีค่าฐานสิบหกเป็น 0x09 โครงสร้างแฟ้มข้อมูลข้อความแสดงดังรูปที่ 2.1



รูปที่ 2.1 แสดงโครงสร้างแฟ้มข้อมูลข้อความ

ตัวอย่างแฟ้มข้อมูลข้อความแสดงเป็นตัวเลขฐานสิบหก แสดงดังรูปที่ 2.2 โดยด้านขวาเป็นตัวอักษรแทนเลขฐานสิบหกเหล่านั้น จะเห็นว่ามีรหัสขึ้นบรรทัดใหม่ (0x0D 0x0A), รหัสตั้งระยะ (0x09) เป็นต้น

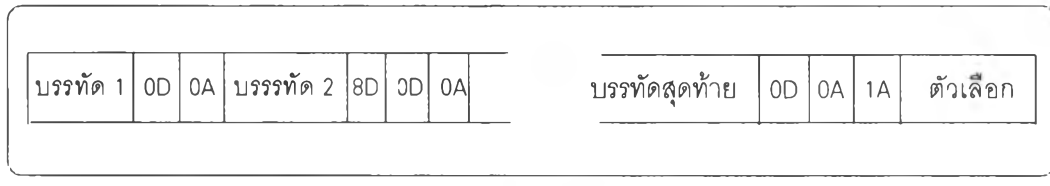
```
43 6F 6E 76 65 72 74 69 - 6F 6E 09 43 6F 6D 70 72  Convertion.Compr
65 73 73 69 6F 6E 09 43 - 77 2D 57 72 69 74 65 72  ession.Cw-Wrier
0D 0A 45 6E 6E 63 72 79 - 70 74 69 6F 6E           .Enncryption
```

รูปที่ 2.2 แสดงตัวอย่างแฟ้มข้อมูลข้อความ

2. โครงสร้างแฟ้มข้อมูลเวิร์ดจุฬา

เวิร์ดจุฬารุ่นที่ 1.52 จะจัดเก็บข้อมูลในแฟ้มข้อมูล เรียงลำดับทีละตัวอักษรตามรหัสสมอ. หรือรหัสเกษตรไปเรื่อยๆจากบรรทัดแรกจนถึงบรรทัดสุดท้าย แต่ละบรรทัดจะถูกคั่นด้วยรหัสขึ้นบรรทัดใหม่ขนาด 2 ไบต์ที่มีค่าเป็นเลขฐานสิบหกคือ 0x0D, 0x0A ถ้าบรรทัดนั้นเกิดจากการจัดบรรทัดที่มีการตัดคำจะมีรหัสตัดคำขนาด 3 ไบต์คือ 0x8D, 0x0D, 0x0A และบรรทัดสุดท้ายของเอกสารจะจบด้วยรหัสจบขนาด 1 ไบต์คือ 0x1A จากนั้นเป็นตัวเลือกต่างๆ ของแฟ้มข้อมูลโครงสร้างแฟ้มข้อมูลเวิร์ดจุฬารุ่นที่ 1.52 แสดงดังรูปที่ 2.3

แต่ละบรรทัดจะเป็นตัวอักษรของข้อความตามรหัสแอสกีโดยเรียงกันไปเรื่อย ข้อความที่มีรูปแบบพิเศษ อาทิตัวเอียง ตัวหนา หรืออื่นๆ จะมีรหัสรูปแบบพิเศษกำกับปิดหัวและท้ายของข้อความนั้น รหัสรูปแบบพิเศษแสดงดังตารางที่ 2.1



รูปที่ 2.3 แสดงโครงสร้างแฟ้มข้อมูลเวิร์ดจุฬา รุ่นที่ 1.5

รูปแบบการพิมพ์	รหัสรูปแบบพิเศษ (ฐานสิบหก)
ตัวปกติ	ไม่มี
ตัวเอียง	0x17
ตัวขีดเส้นใต้ 1 เส้น	0x13
ตัวขีดเส้นใต้ 2 เส้น	0x12
ตัวหนา	0x02
ตัวใหญ่	0x05
ตัวยก	0x14
ตัวห้อย	0x16

ตารางที่ 2.1 แสดงรหัสรูปแบบพิเศษ

เวิร์ดจุฬาสามารถสร้างตารางในเอกสารได้ โดยเวิร์ดจุฬาได้เตรียมอักษรกราฟฟิกไว้ตีเส้นส่วนประกอบของตาราง ดังนั้นบรรทัดที่มีเส้นส่วนประกอบของตารางจะมีรหัสตารางนั้นแทรกอยู่ รหัสตารางแสดงดังตารางที่ 2.2

ตัวอย่างแฟ้มข้อมูลเวิร์ดจุฬา รุ่นที่ 1.52 ดังรูปที่ 2.4 โดยแสดงเป็นตัวเลขฐานสิบหก จะเห็นว่ามีรหัสขึ้นบรรทัดใหม่ (0x0D 0x0A), รหัสตัดคำ (0x8D 0x0D 0x0A), รหัสรูปแบบพิเศษแบบกำกับปิดหัวท้ายข้อความรูปแบบตัวหนา (0x02), รหัสรูปแบบพิเศษแบบกำกับปิดหัวท้ายข้อความรูปแบบตัวใหญ่ (0x05) และรหัสจบ (0x1A) จากนั้นเป็นตัวเลือกต่างๆของแฟ้มข้อมูล

เส้นส่วนประกอบของตาราง	รหัสตาราง (ฐานสิบหก)
┌	0x98
┬	0x91
┐	0x99
├	0x93
│	0x96
┤	0x92
└	0x9A
┴	0x90
┘	0x9B
─	0x95
┼	0x8F

ตารางที่ 2.2 แสดงรหัสตาราง

```
6E 6F 72 6D 61 6C 8D 0D - 0A 17 69 74 61 6C 69 63  normal.. italic
61 72 67 65 05 1A 4F 50 - 54 20 31 2E 35 30 01 00  large..OPT 1.50..
17 8D 0D 0A 02 62 6F 6C - 64 02 0D 0A 05 65 6E 6C  .....bold.... enl
00 00 00 00 00 00 00 00 - 00 00 04 21 08 42 10 84  ..........!.B..
21 08 40 00 00 00 00 00 - 00 00 00 00 00 00 00 00  !.@.....  .....
01 00 00 00 01 00 00 00 - 01 00 00 00 01 00 00 00  ................
01 00 00 00 01 00 4B 00 - 1E 00 01 00 04 00 4E 00  ......K.......N.
13 00 6E 00                                        ..n
```

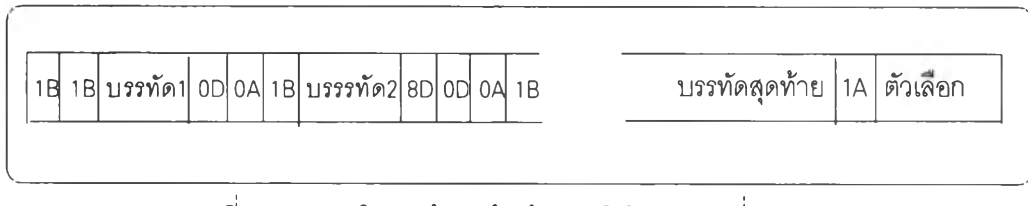
รูปที่ 2.4 แสดงตัวอย่างแฟ้มข้อมูลเวิร์ดจุฬา รุ่นที่ 1.52

เวิร์ดจุฬารุ่นที่ 1.6 จัดเก็บข้อมูลในแฟ้มข้อมูลเรียงลำดับไปเรื่อยๆจากบรรทัดแรกจนถึงบรรทัดสุดท้ายเช่นกัน เวิร์ดจุฬารุ่นที่ 1.6 ปรับปรุงให้ใช้แบบอักษรได้ 4 แบบ ข้อความที่ใช้แบบอักษร 2, 3 และ 4 จะมีรหัสแบบอักษรกำกับปิดหัวท้าย แต่ข้อความที่ใช้แบบอักษร 1 จะไม่มีรหัสแบบอักษรกำกับปิดหัวท้าย รหัสแบบอักษรแสดงดังตารางที่ 2.3

แบบอักษร	รหัสแบบอักษร (ฐานสิบหก)
แบบอักษร 1	0x1B
แบบอักษร 2	0x1C
แบบอักษร 3	0x1D
แบบอักษร 4	0x1E

ตารางที่ 2.3 แสดงรหัสแบบอักษร

บรรทัดแรกจะขึ้นต้นด้วยรหัสแบบอักษร 1(0x1B) จำนวน 2 ไบต์ แต่ละบรรทัดจะถูกคั่นด้วยรหัสขึ้นบรรทัดใหม่ขนาด 2 ไบต์คือ 0x0D, 0x0A และตามด้วยรหัสแบบอักษร 1 (0x1B) ทุกบรรทัด ถ้าบรรทัดนั้นเกิดจากการจัดบรรทัดที่มีการตัดคำจะมีรหัสตัดคำขนาด 3 ไบต์คือ 0x8D, 0x0D, 0x0A และตามด้วยรหัสแบบอักษร 1 (0x1B) เมื่อเกิดการตัดคำจะใช้รหัสแทรก 0xA0 แทนช่องว่าง (0x20) และบรรทัดสุดท้ายของเอกสารจะจบด้วยรหัสจบขนาด 1 ไบต์คือ 0x1A จากนั้นเป็นตัวเลือกต่างๆ ของแฟ้มข้อมูล โครงสร้างแฟ้มข้อมูลเวิร์ดจุฬารุ่นที่ 1.6 แสดงดังรูปที่ 2.5



รูปที่ 2.5 แสดงโครงสร้างแฟ้มข้อมูลเวิร์ดจุฬารุ่นที่ 1.6

เวิร์ดจุฬารุ่นที่ 1.6 สามารถพิมพ์สมการคณิตศาสตร์ได้ โดยพิมพ์คอลัมน์แรกด้วยคำสั่งจุด ".M" ตามด้วยข้อความที่อธิบายการเขียนสมการคณิตศาสตร์แสดงดังตารางที่ 2.4

ตัวอย่างแฟ้มข้อมูลเวิร์ดจุฬารุ่นที่ 1.6 ดังรูปที่ 2.6 โดยแสดงเป็นตัวเลขฐานสิบหก จะเห็นว่ามีรหัสขึ้นบรรทัดใหม่ (0x0D 0x0A), รหัสแบบอักษร 1 (0x1B), รหัสแทรกจากการตัดคำ (0xA0), รหัสตัดคำ (0x8D 0x0D 0x0A), รหัสแบบอักษร 2 กำกับปิดหัวท้ายข้อความ (0x1C) และรหัสจบ (0x1A) จากนั้นเป็นตัวเลือกต่างๆ

การเขียนสมการ	หมายเหตุ
.M<ข้อความ> .M<ข้อความ1><ตัวเชื่อม><ข้อความ2>	<ข้อความ> ใส่ในเครื่องหมายคำพูดหรือไม่ก็ได้ <ตัวเชื่อม> ที่เป็นไปได้ มีดังนี้ 1. ^ พิมพ์ในลักษณะยกกำลัง 2. / พิมพ์ในลักษณะเศษส่วน 3. ~ ตัวเชื่อมข้อความ
.M<function>(<ข้อความ1>,<ข้อความ2>,...)	<function> ที่เป็นไปได้ มีดังนี้ int(a,b,c), cint(a,b,c), lim(a.b), root(x), conj(x), abs(x), sum(x), pie(x)

ตารางที่ 2.4 แสดงการเขียนสมการคณิตศาสตร์

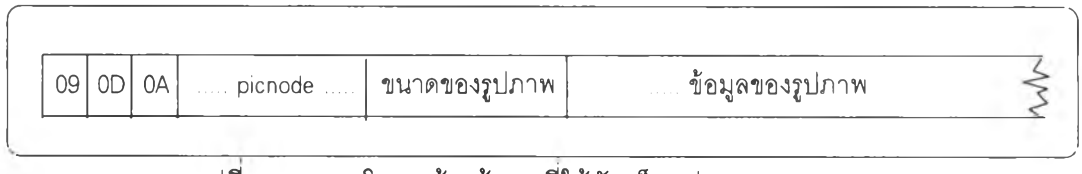
```
1B 1B 66 6F 6E 74 31 A0 - A0 A0 A0 A0 A0 A0 A0 A0   ..font1.........
20 1C 66 6F 6E 74 32 1C - 8D 0D 0A 1B 1D 66 6F 6E    font2.....fon
74 33 20 1D 1E 66 6F 6E - 74 34 1E 1A 4F 50 54 20   t3 ..font4 OPT
31 2E 35 30 01 00 00 0C - 00 00 00 00 00 00 00 00   1.50............
04 21 08 42 10 84 21 08 - 00 00 00 00 00 00 00 00   .!.B..!.........
00 00 01 00 00 00 01 00 - 00 00 01 00 14 00 21 00   ..............!.
01 00 04 00 4E 00 13 00 - 6E 00                     ....N...n.
```

รูปที่ 2.6 แสดงตัวอย่างแฟ้มข้อมูลเวิร์ดจุฬารุ่นที่ 1.6

3. <u>โครงสร้างแฟ้มข้อมูลจุฬาจารึก 77</u>

โครงสร้างแฟ้มข้อมูลจุฬาจารึก 77 มีลักษณะเช่นเดียวกับโครงสร้างแฟ้มข้อมูลเวิร์ดจุฬารุ่นที่ 1.52 กล่าวคือจัดเก็บข้อมูลเรียงลำดับไปเรื่อยๆจากบรรทัดแรกจนถึงบรรทัดสุดท้าย แต่ละบรรทัดจะมีรหัสขึ้นบรรทัดใหม่ (0x0D 0x0A) บรรทัดที่เกิดจากตัดคำจะมีรหัสตัดคำ (0x8D 0x0D 0x0A) จบเอกสารด้วยรหัสจบ (0x1A) ข้อความที่มีรูปแบบพิเศษจะมีรหัสรูปแบบพิเศษกำกับปิดหัวและท้ายข้อความ มีรหัสตารางแทนตัวอักษรกราฟฟิกของเส้นส่วนประกอบของตาราง จุฬาจารึกได้เปลี่ยนรหัสแทรกที่เกิดจากการตัดคำโดยใช้รหัสแทรก 0xA0 แทนช่องว่าง (0x20) นอกจากนี้จุฬาจารึก 77 ได้เพิ่มความสามารถให้นำรูปภาพเข้ามาประกอบในเอกสารได้ โดยกำหนดให้ใช้รูปภาพได้ 2 รูปแบบ แบบแรกคือแบบแผนที่บิต (bitmap - BMP) เป็นรูปภาพที่ไม่ขึ้น

กับอุปกรณ์แสดงผล โดยจัดเก็บการกำหนดจุดต่างๆที่ประกอบกันเป็นรูปภาพและจานสีที่รูปภาพใช้ แบบสองคือแบบเมตาไฟล์ (metafile - WMF) เป็นรูปภาพที่จัดเก็บตามมาตราฐานของวินโดว์ โดยเป็นชุดคำสั่งทางกราฟฟิกที่ประกอบกันเป็นรูปภาพ ในที่นี้จะกล่าวเฉพาะโครงสร้างข้อมูลที่ใช้จัดเก็บรูปภาพ จะมีรหัสรูปภาพมีขนาด 3 ไบต์คือ 0x09 0x0D 0x0A นำหน้าเพื่อบอกว่าข้อมูลที่ตามหลังนี้เป็นข้อมูลของรูปภาพ โดยเริ่มจากโครงสร้างข้อมูล picnode ตามด้วยขนาดของรูปภาพและข้อมูลของรูปภาพ (คณะวิจัยจุฬาจารึก, 2538) โครงสร้างข้อมูลที่ใช้จัดเก็บรูปภาพ แสดงดังรูปที่ 2.7



รูปที่ 2.7 แสดงโครงสร้างข้อมูลที่ใช้จัดเก็บรูปภาพ

3.1 โครงสร้างข้อมูล picnode ประกอบด้วยข้อมูลดังนี้

```
int             x;          // เป็นตำแหน่งของคอลัมน์เริ่มต้นของรูปภาพ
int             dx;         // เป็นความกว้างของรูปภาพ มีหน่วยเป็นคอลัมน์
                              โดยจัดเก็บในหน่วยของหนึ่งร้อย
int             dy;         // เป็นความสูงของรูป มีหน่วยเป็นบรรทัด
                              โดยจัดเก็บในหน่วยของหนึ่งร้อย
int             mx;         // เป็นร้อยละของการย่อ/ขยายรูปในแนวแกน x
int             my;         // เป็นร้อยละของการย่อ/ขยายรูปในแนวแกน y
BYTE            reserved;   // สำรองไว้ใช้ในอนาคต
METAFILEPICT    mfpic;      // เป็นโครงสร้างข้อมูลรูปภาพแบบ WMF
                              ถ้ารูปภาพแบบ BMP ค่า hMF ของ mfpic จะเป็น null
                              (ดูเพิ่มเติม 3.1.1)
int             hdibmp;     // เป็นค่าแฮนเดิลของรูปภาพแบบ BMP
                              ถ้ารูปภาพแบบ WMF ค่าของ hdibmp จะเป็น null
```

3.1.1 โครงสร้างข้อมูลของ mfpic ประกอบด้วยข้อมูลดังนี้

```
int         mm;     // เป็นค่าที่บอกถึง mapping mode ของรูปภาพ WMF
int         xExt;   // เป็นค่าความกว้างเดิมของรูปภาพ WMF
int         yExt;   // เป็นค่าความสูงเดิมของรูปภาพ WMF
HMETAFILE   hMF;    // เป็นค่าแฮนเดิลของรูปภาพแบบเมตาไฟล์
                       ถ้ารูปภาพแบบ BMP ค่านี้จะเป็น null
```

ตัวอย่างแฟ้มข้อมูลจุฬาจารึก 77 ที่มีรูปภาพแบบ WMF ในเอกสาร ดังรูปที่ 2.8 แสดงเป็นตัวเลขฐานสิบหก จะเห็นว่ามีรหัสรูปภาพ (0x09 0x0D 0x0A) ตามด้วยโครงสร้างข้อมูล picnode ซึ่งค่าของ hdibmp จะเป็น null ตามด้วยขนาดรูปภาพ และบางส่วนของข้อมูลรูปภาพ

```
57 4D 46 0D 0A 09 0D 0A  - 00 00 AB 03 77 01 64 00   WMF.........w.d.
64 00 FF 08 00 C5 07 C5  - 07 66 10 56 00 00 DC 07   d........f.V....
00 00 01 00 09 00 00 03  - EE 03 00 00 04 00 8E 01   ....  ..........
00 00 00 00 05 00 00 00  - 0B 02 8B 1C 99 0C 05 00   ..... ... ... ...
00 00 0C 02 10 03 10 03  - 1D 00 00 00 26 06 0F 00   ............&...
2F 00 4D 47 58 49 3A 20  - 63 72 65 61 74 65 64 20   /.MGXI: created
62 79 20 4D 69 63 72 6F  - 67 72 61 66 78 2C 20 49   by Micrografx, I
6E 63 2E 20 4D 67 78 67  - 72 65 20 76 31 2E 30 30   nc. Mgxgre v1.00
00 00 04 00 00 00 03 01  - 08 00 05 00 00 00 0B 02   ................
8B 1C 99 0C 05 00 00 00  - 0C 02 10 03 10 03 07 00   ................
```

รูปที่ 2.8 แสดงบางส่วนของแฟ้มข้อมูลจุฬาจารึก 77 ที่มีรูปภาพแบบ WMF

เนื่องจากการแปลงโครงสร้างแฟ้มข้อมูลส่วนรูปภาพจำเป็นต้องใช้ค่าความกว้างเดิมและค่าความยาวเดิมของรูปภาพ โดยค่าความกว้างเดิมของรูปภาพแบบ WMF (xExt) และค่าความสูงเดิมของรูปภาพแบบ WMF (yExt) จัดเก็บใน mfpic ในโครงสร้างข้อมูล picnode ในขณะที่ค่าความกว้างเดิมของรูปภาพแบบ BMP (biWidth) และค่าความสูงเดิมของรูปภาพแบบ BMP (biHeight) จัดเก็บใน bmiHeader ซึ่งเป็นข้อมูลส่วนแรกของข้อมูลรูปภาพ

3.2 โครงสร้างข้อมูล bmiHeader ประกอบด้วยข้อมูลดังนี้

```
DWORD       biSize;          // จำนวนไบต์ของโครงสร้างข้อมูล bmiHeade
LONG        biWidth;         // ความกว้างเดิมของรูปภาพแบบ BMP
LONG        biHeight;        // ความสูงเดิมของรูปภาพแบบ BMP
WORD        biPlanes;        // จำนวนระนาบของอุปกรณ์แสดงผล
WORD        biBitCount;      // จำนวนบิตที่ใช้ต่อหนึ่งจุดภาพ
DWORD       biCompression;   // รูปแบบการอัดข้อมูลรูปภาพ
DWORD       biSizeImage;     // จำนวนไบต์ของรูปภาพ
LONG        biXpelsPerMeter; // ค่าความละเอียดในแนวนอนของอุปกรณ์แสดงผล
LONG        biYpelsPerMeter; // ค่าความละเอียดในแนวตั้งของอุปกรณ์แสดงผล
DWORD       biClrUsed;       // จำนวนดรรชนีสีที่ใช้สำหรับรูปภาพ
DWORD       biClrImportant;  // จำนวนดรรชนีสีที่จำเป็นสำหรับรูปภาพ
```

ตัวอย่างแฟ้มข้อมูลจุฬาจารึก 77 ที่มีรูปภาพแบบ BMP ในเอกสาร ดังรูปที่ 2.9 แสดงเป็นตัวเลขฐานสิบหก จะเห็นว่ามีรหัสรูปภาพ (0x09 0x0D 0x0A) ตามด้วยโครงสร้างข้อมูล picnode ซึ่งค่า hMF ของ mfpic จะเป็น null ตามด้วยขนาดรูปภาพ และบางส่วนของข้อมูลรูปภาพแบบ BMP ซึ่งมีโครงสร้างข้อมูล bmiHeader อยู่ในส่วนแรกของข้อมูลรูปภาพ

```
42 4D 50 0D 0A 09 0D 0A  - 00 00 90 01 A0 00 64 00   BMP.....d
64 00 76 FE  FF 76 FC 6A  - 00 00 00 08 06 09 40 08   d v  v j.....@
00 00 28 00  00 00 20 00 -  00 00 20 00 00 00 01 00   ..(. ... .....
08 00 00 00 00 00 00 00  -  00 00 00 00 00 00 00 00   ................
00 00 00 01 00 00 00 00  -  00 00 00 00 00 00 00 00   ................
BF 00 00 BF 00 00 00 BF -  BF 00 BF 00 00 00 BF 00   ..... ........
BF 00 BF BF 00 00 C0 C0  - C0 00 C0 DC C0 00 F0 CA   ................
A6 00 00 00 00 00 00 00  -  00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00  -  00 00 00 00 00 00 00 00   ... . ... .....
00 00 00 00 00 00 00 00  -  00 00 00 00 00 00 00 00   ... . ... .....
```

รูปที่ 2.9 แสดงบางส่วนของแฟ้มข้อมูลจุฬาจารึก 77 ที่มีรูปภาพแบบ BMP

4. โครงสร้างแฟ้มข้อมูลจุฬาจารึก 78

จุฬาจารึก 78 ได้ออกแบบโครงสร้างแฟ้มข้อมูลให้จัดเก็บข้อมูลอย่างมีระเบียบมากขึ้น ไม่จัดเก็บข้อมูลเรียงตามลำดับไปเรื่อยๆจากบรรทัดแรกจนถึงบรรทัดสุดท้าย แต่จัดเก็บข้อมูลเป็นส่วนต่างๆ 10 ส่วนคือ ส่วน Header, ส่วน Style, ส่วน Section, ส่วน Font, ส่วน Picture, ส่วน Text, ส่วน Index, ส่วน Attribute, ส่วน Overlay Drawing และส่วน TTF Embed แต่ละส่วนมีรายละเอียดต่อไปนี้ (คณะวิจัยจุฬาจารึก, 2538)

4.1 ส่วน Header สำหรับตรวจสอบว่าแฟ้มข้อมูลนั้นเป็นจุฬาจารึก 78 หรือไม่ และโครงสร้างดังกล่าวถูกต้องหรือไม่ โดยมีตัวระบุแฟ้มข้อมูลจุฬาจารึก 78 จุดเริ่มต้นของแต่ละส่วนในแฟ้มข้อมูล และอื่นๆ ส่วน Header มีโครงสร้างข้อมูลดังนี้

```
DWORD       dwIdent;        // ตัวระบุจุฬาจารึก 78 มีค่า 0xA8CC4E00
WORD        dwSize;         // ขนาดของส่วน Header
DWORD       pgStyle;        // จุดเริ่มต้นของส่วน Style
DWORD       pgSection;      // จุดเริ่มต้นของส่วน Section
DWORD       pgText;         // จุดเริ่มต้นของส่วน Text
DWORD       pgIdx;          // จุดเริ่มต้นของส่วน Index
DWORD       pgAttr;         // จุดเริ่มต้นของส่วน Attribute
DWORD       pgFont;         // จุดเริ่มต้นของส่วน Font
DWORD       pgPicture;      // จุดเริ่มต้นของส่วน Picture
DWORD       pgDrawing;      // จุดเริ่มต้นของส่วน Overlay Drawing
DWORD       pgTTFEmbed;     // จุดเริ่มต้นของส่วน TTF Embed
int         ttfembed;       // แฟล็กของคุณสมบัติ TTF Embed
int         zoom;           // ค่าการขยายขนาด หน่วยเป็นเปอร์เซนต์
int         fitzoom;        // ภาวะการขยาย Page Width หรือ Whole Page
int         deftab,         // ค่าตั้งระยะเริ่มต้น (default tab)
int         viewallchar;    // ภาวะแสดงอักขระทุกตัว
int         autosavemin;    // เวลาจัดเก็บเอกสารอัตโนมัติ หน่วยเป็นนาที
int         compressencrypt;// แฟล็กของคุณสมบัติอัดข้อมูลหรือเข้ารหัสลับ
```

```
int     bReadonly;                  // ภาวะเปิดเอกสารสำหรับอ่านเท่านั้น
char    title[MAXDOCINFO];          // ชื่อเรื่อง โดย MAXDOCINFO เท่ากับ 81
char    Subject[MAXDOCINFO];        // หัวข้อเรื่อง โดย MAXDOCINFO เท่ากับ 81
char    Author[MAXDOCINFO];         // ผู้เขียน โดย MAXDOCINFO เท่ากับ 81
char    Keywords[MAXDOCINFO];       // คำหลัก โดย MAXDOCINFO เท่ากับ 81
char    Password[MAXPASSWORD];      // รหัสผ่าน โดย MAXPASSWORD เท่ากับ 9
int     biSnapTo;                   // แฟล็กของ Snap
int     xGrid;                      // โครงข่ายแกน x หน่วยเป็น twip
int     yGrid;                      // โครงข่ายแกน y หน่วยเป็น twip
DWORD   unused1;
DWORD   unused2;
DWORD   unused3;
DWORD   unused4;
```

4.2 ส่วน Style เป็นส่วนสำหรับจัดเก็บสไตล์ (style) ที่แตกต่างกัน สามารถกำหนดสไตล์ที่ใช้เป็นประจำในเอกสารเช่น แบบอักษร ขนาดตัวอักษร รูปแบบพิเศษของข้อความ ย่อหน้า เป็นต้น เพื่อความสะดวกในการใช้งาน นอกจากนี้ยังมีตัวระบุส่วน Style ใช้สำหรับตรวจสอบโครงสร้างแฟ้มข้อมูลจุฬาจารึก ส่วน Style มีโครงสร้างข้อมูลดังนี้

```
WORD        id;         // ตัวระบุส่วน Style มีค่า 0x014E
DWORD       size;       // ขนาดของส่วน Style
DWORD       numstyle;   // จำนวนสไตล์
STYLEINFO   styleinfo[ ];   // แถวลำดับรายละเอียดของสไตล์ (ดูรายละเอียด 4.2.1)
```

4.2.1 STYLEINFO เป็นรายละเอียดของสไตล์ มีโครงสร้างข้อมูลดังนี้

```
char    stylename[MAXSTYLENAME]; // ชื่อสไตล์ โดย MAXSTYLENAME เท่ากับ20
int     leftindent;     // ย่อหน้าด้านซ้าย หน่วยเป็น twip
int     rightindent;    // ย่อหน้าด้านขวา หน่วยเป็น twip
int     firstindent;    // ย่อหน้าบรรทัดแรก หน่วยเป็น twip
```

```
int         spacebefore;    // ระยะห่างด้านบน หน่วยเป็น point
int         spaceafter;     // ระยะห่างด้านล่าง หน่วยเป็น point
int         linespacing;    // ระยะห่างของบรรทัด หน่วยเป็น point
BYTE        alignment;      // วางแนวชิดซ้าย, กึ่งกลาง, ชิดขวา, ตามกำหนด
TABINFO     tabinf[MAXTAB]; // ตั้งระยะ โดย MAXTAB เท่ากับ 10 (ดูรายละอียด 4.2.2)
BYTE        textcolor;      // สีของข้อความ
BYTE        textattribute;  // attribute ของข้อความ
BYTE        floating;       // ถ้าเป็น 1 จะพิมพ์ย่อหน้านั้นในหน้ากระดาษเดียวกัน
BYTE        bullet;         // ถ้าเป็น 0 ไม่มี bullet, 1 มี bullet
BYTE        bulletcode;     // รหัสแอสกีของ bullet
BYTE        bulletcolor;    // สีของ bullet
int         bullettotext;   // ระยะห่างของ bullet กับข้อความ หน่วยเป็น twip
int         indenttotext;   // ยังไม่พัฒนา
WORD        bulletfontidx;  // ดรรชนีแบบอักษรของ bullet
UINT        fontindex;      // ดรรชนีแบบอักษรของข้อความ
FONTSTYLEINFO  textfont;    // รายละเอียดแบบอักษรของสไตล์ที่ข้อความใช้
                            (ดูเพิ่มเติม 4.2.3)
FONTSTYLEINFO  bulletfont;  // รายละเอียดแบบอักษรของสไตล์ที่ bullet ใช้
                            (ดูเพิ่มเติม 4.2.3)
```

4.2.2 TABINFO เป็นรายละเอียดของการตั้งระยะ มีโครงสร้างข้อมูลดังนี้

```
BYTE        type,           // ถ้าเป็น 0 ด้านซ้าย, 1 กึ่งกลาง, 2 ด้านขวา, 3 decimal
int         tabpos;         // ตำแหน่งแต่ละตั้งระยะ หน่วยเป็น twip
```

4.2.3 FONTSTYLEINFO เป็นรายละเอียดแบบอักษรของสไตล์ มีโครงสร้างข้อมูลดังนี้

```
char        fontname[LF_FACESIZE]; // ชื่อแบบอักษร
int         point;                 // ขนาดตัวอักษร หน่วยเป็น point
```

```
BYTE        bold;           // รูปแบบตัวหนา
BYTE        italic;         // รูปแบบตัวเอียง
int         npoint;         // ขนาดตัวอักษรที่มีรูปแบบตัวยก, ตัวห้อย
```

4.3 ส่วน Section เป็นส่วนที่จัดเก็บข้อมูลเกี่ยวกับการจัดหน้ากระดาษต่างๆ เช่น ขนาดกระดาษ ขอบเผื่อ (margin) การวางหน้ากระดาษ ลักษณะของย่อหน้า เป็นต้น นอกจากนี้ยังมีตัวระบุส่วน Section ใช้สำหรับตรวจสอบโครงสร้างแฟ้มข้อมูลจุฬาจารึก ส่วน Section มีโครงสร้างข้อมูลดังนี้

```
WORD        id;             // ตัวระบุส่วน Section มีค่า 0x024E
DWORD       size;           // ขนาดของส่วน Section
DWORD       numsection;     // จำนวน Section
WORD        papertype;      // ชนิดกระดาษ (ดูเพิ่มเติมใน window)
WORD        papersource;    // แหล่งป้อนกระดาษ (ดูเพิ่มเติมใน window)
WORD        paperwidth;     // ความกว้างของกระดาษ หน่วยเป็น twip
WORD        paperheight;    // ความสูงของกระดาษ หน่วยเป็น twip
WORD        orientation;    // การวางหน้ากระดาษ
WORD        leftmargin;     // ขอบเผื่อด้านซ้าย หน่วยเป็น twip
WORD        rightmargin;    // ขอบเผื่อด้านขวา หน่วยเป็น twip
WORD        topmargin;      // ขอบเผื่อด้านบน หน่วยเป็น twip
WORD        bottommargin;   // ขอบเผื่อด้านล่าง หน่วยเป็น twip
WORD        headermargin;   // ขอบเผื่อของหัวกระดาษ หน่วยเป็น twip
WORD        footermargin;   // ขอบเผื่อของท้ายกระดาษ หน่วยเป็น twip
int         pagewidth;      // ความกว้างของหน้า หน่วยเป็น twip
int         pageheight;     // ความสูงของหน้า หน่วยเป็น twip
int         pageno;         // ยังไม่พัฒนา
int         startpage;      // หมายเลขหน้าเริ่มต้น
BYTE        pagetype;       // การพิมพ์หน้ากระดาษ
int         pagexpos;       // ยังไม่พัฒนา
int         pageypos;       // ยังไม่พัฒนา
```

```
char            pagetitle[MAXPAGETITLE];  // หัวกระดาษโดย MAXPAGETITLE เป็น 20
BYTE            pagelang;                 // ถ้าเป็น 0 พิมพ์เลขอารบิค, 1 พิมพ์เลขไทย
UINT            pagestyleidx;             // ดรรชนีสไตล์ของหัวกระดาษ
DWORD           numheaderpara;            // จำนวนย่อหน้าของหัวกระดาษ
PARAGRAPHINFO   headerparagraphinfo[ ];// แถวลำดับย่อหน้าของหัวกระดาษ
                                          (ดูเพิ่มเติม 4.3.1)
DWORD           numbodypara;              // จำนวนย่อหน้าของเนื้อเรื่อง
PARAGRAPHINFO   bodyparagraphinfo[ ];  // แถวลำดับย่อหน้าของเนื้อเรื่อง
                                          (ดูเพิ่มเติม 4.3.1)
DWORD           numfooterpara;            // จำนวนย่อหน้าของท้ายกระดาษ
PARAGRAPHINFO   footerparagraphinfo[ ]; // แถวลำดับย่อหน้าของท้ายกระดาษ
                                          (ดูเพิ่มเติม 4.3.1)
```

4.3.1 PARAGRAPHINFO เป็นรายละเอียดของย่อหน้า มีโครงสร้างข้อมูลดังนี้

```
int         leftindent;     // ย่อหน้าด้านซ้าย หน่วยเป็น twip
int         rightindent;    // ย่อหน้าด้านขวา หน่วยเป็น twip
int         firstindent;    // ย่อหน้าบรรทัดแรก หน่วยเป็น twip
int         spacebefore;    // ระยะห่างด้านบน หน่วยเป็น point
int         spaceafter;     // ระยะห่างด้านล่าง หน่วยเป็น point
int         linespacing;    // ระยะห่างของบรรทัด หน่วยเป็น point
BYTE        bullet,         // ถ้าเป็น 0 ไม่มี bullet, 1 มี bullet
BYTE        bulletcode;     // รหัสแอสกีของ bullet
BYTE        bulletcolor;    // สีของ bullet
WORD        bulletfontidx;  // ดรรชนีแบบอักษรของ bullet
int         bullettotext;   // ระยะห่างของ bullet กับข้อความ หน่วยเป็น twip
int         indenttotext;   // ยังไม่พัฒนา
BYTE        alignment;      // วางแนวชิดซ้าย, กึ่งกลาง, ชิดขวา, ตามกำหนด
TABINFO     tabinf[MAXTAB]; // ตั้งระยะ โดย MAXTAB เท่ากับ 10 (ดูเพิ่มเติม 4.2.2)
```

```
BYTE        textcolor;      // สีของข้อความ
BYTE        textattribute;  // attribute ของข้อความ
BYTE        floating;       // ถ้าเป็น 1 จะพิมพ์ย่อหน้านั้นในหน้ากระดาษเดียวกัน
UINT        styleidx;       // ดรรชนีสไตล์ของย่อหน้า
UINT        fontindex;      // ดรรชนีแบบอักษรของข้อความ
WORD        numdrawinfo;    // จำนวนรูปวาดในย่อหน้านี้
DWORD       unused1;
DWORD       unused2;
```

4.4 ส่วน Font ส่วนนี้จัดเก็บแบบอักษรที่ใช้ในเอกสาร และมีตัวระบุส่วน Font ใช้สำหรับตรวจสอบโครงสร้างแฟ้มข้อมูลจุฬาจารึก ส่วน Font มีโครงสร้างข้อมูลดังนี้

```
WORD    id;             // ตัวระบุส่วน Font มีค่า 0x034E
DWORD   size;           // ขนาดส่วน Font
DWORD   numfont;        // จำนวนแบบอักษรที่ใช้ในเอกสาร
FONTINFO fontinfo[ ];   // แถวลำดับรายละเอียดแบบอักษร (ดูเพิ่มเติม 4.4.1)
```

4.4.1 FONTINFO เป็นรายละเอียดของแบบอักษร มีโครงสร้างข้อมูลดังนี้

```
char        fontname[LF_FACESIZE]; // ชื่อแบบอักษร
int         point;          // ขนาดตัวอักษร หน่วยเป็น point
BYTE        bold;           // รูปแบบตัวหนา
BYTE        italic;         // รูปแบบตัวเอียง
int         npoint;         // ขนาดตัวอักษรที่มีรูปแบบตัวยก, ตัวห้อย
BYTE        pitchandfamily;
BYTE        owner;
int         suboffset;      // ระยะห่างของตัวห้อยจากบรรทัด
int         superoffset;    // ระยะห่างของตัวยกจากบรรทัด
```

4.5 ส่วน Picture ส่วนนี้จัดเก็บรูปภาพที่นำมาประกอบในเอกสาร และมีตัวระบุส่วน picture ใช้สำหรับตรวจสอบโครงสร้างแฟ้มข้อมูลจุฬาจารึก มีโครงสร้างข้อมูลดังนี้

```
WORD        id;           // ตัวระบุส่วน Picture มีค่า 0x144E
DWORD       size;         // ขนาดส่วน Picture
DWORD       numpicture;   // จำนวนรูปภาพ
PICTUREINFO pictureinfo[ ]; // แถวลำดับรายละเอียดรูปภาพ (ดูเพิ่มเติม 4.5.1)
```

4.5.1 PICTUREINFO เป็นรายละเอียดของรูปภาพ มีโครงสร้างข้อมูลดังนี้

```
int         width;        // ความกว้างของรูปภาพ หน่วยเป็น twip
int         height;       // ความสูงของรูปภาพ หน่วยเป็น twip
int         orgwidth;     // ความกว้างเดิมของรูปภาพ หน่วยเป็น twip
int         orgheight;    // ความสูงเดิมของรูปภาพ หน่วยเป็น twip
int         vertadj;      // ระยะห่างในแนวนอน หน่วยเป็น twip
BYTE        type;         // ถ้าเป็น 1 = WMF, 2 = DIB, 3 = OLE
int         scalex;       // สัดส่วนแนวแกน x ของรูปภาพ
int         scaley;       // สัดส่วนแนวแกน y ของรูปภาพ
int         cropleft;     // ตัดรูปภาพด้านซ้าย
int         croptop;      // ตัดรูปภาพด้านบน
int         cropright;    // ตัดรูปภาพด้านขวา
int         cropbottom;   // ตัดรูปภาพด้านล่าง
DWORD       picturesize;  // ขนาดของรูปภาพ หน่วยเป็นไบต์
BYTE        hdata;
```

4.6 ส่วน Text ส่วนนี้จัดเก็บข้อความในเอกสารและรหัสรูปภาพ (0x10) แทนรูปภาพที่นำมาประกอบในเอกสาร โดยข้อมูลเกี่ยวกับรูปภาพจัดเก็บในส่วน Picture นอกจากนี้มีตัวระบุส่วน Text ใช้สำหรับตรวจสอบโครงสร้างแฟ้มข้อมูลจุฬาจารึก มีโครงสร้างข้อมูลดังนี้

```
WORD        id;           // ตัวระบุส่วน Text มีค่า 0x054E
DWORD       size;         // ขนาดของส่วน Text
```

```
BYTE        text[ ];        // แถวลำดับรหัสแอสกีของข้อความ และรหัสอื่นดังนี้
                            0x00 รหัสจบข้อความ, 0x09 รหัสตั้งระยะ,
                            0x10 รหัสรูปภาพ,    0x0A รหัสตัดคำด้วยมือ,
                            0x0C รหัส pagebreak ด้วยมือ, 0x0D รหัสจบย่อหน้า
```

4.7 ส่วน Index ส่วนนี้จัดเก็บดรรชนี (index) แบบอักษรของข้อความ โดยข้อมูลเกี่ยวกับแบบอักษรจัดเก็บในส่วน Font หรือดรรชนีรูปภาพโดยข้อมูลเกี่ยวกับรูปภาพจัดเก็บในส่วน Picture นอกจากนี้มีตัวระบุส่วน Index ใช้สำหรับตรวจสอบโครงสร้างแฟ้มข้อมูลจุฬาจารึก ส่วน Index มีโครงสร้างข้อมูลดังนี้

```
WORD        id;             // ตัวระบุส่วน Index มีค่า 0x064E
DWORD       size;           // ขนาดของส่วน Index
INDEXINFO   indexinfo[ ];   // แถวลำดับรายละเอียดของดรรชนี (ดูเพิ่มเติม 4.7.1)
```

4.7.1 INDEXINFO เป็นรายละเอียดของดรรชนี มีโครงสร้างข้อมูลดังนี้

```
WORD        index;          // ดรรชนีแบบอักษรของข้อความ
WORD        numusedindex; // จำนวนตัวอักษรที่ใช้ดรรชนี
```

4.8 ส่วน Attribute ส่วนนี้จัดเก็บลักษณะประจำ (attribute) และสีของข้อความ และมีตัวระบุส่วน Attribute ใช้สำหรับตรวจสอบโครงสร้างแฟ้มข้อมูลจุฬาจารึก ส่วน Attribute มีโครงสร้างข้อมูลดังนี้

```
WORD            id;             // ตัวระบุส่วน Attribute มีค่า 0x074E
DWORD           size;           // ขนาดของส่วน Attribute
ATTRIBUTEINFO   attributeinfo[ ]; // รายละเอียดลักษณะประจำและสีของข้อความ
                                  (ดูเพิ่มเติม 4.8.1)
```

4.8.1 ATTRIBUTEINFO มีโครงสร้างข้อมูลดังนี้

```
BYTE        attribute;      // ถ้าบิต 4-7 เป็น 0x00 ตัวปกติ,
                            0x10 ตัวห้อย, 0x20 ตัวยก, 0x40 ขีดเส้นใต้ 1 เส้น,
```

0x80 ขีดเส้นใต้ 2 เส้น

บิต 0-3 กำหนดสีข้อความ 16 สี

```
WORD        numusedattribute;  // จำนวนตัวอักษรที่ใช้ลักษณะประจำนี้
```

4.9 ส่วน Overlay Drawing ส่วนนี้จัดเก็บข้อมูลของรูปวาด และมีตัวระบุส่วน Overlay Drawing ใช้สำหรับตรวจสอบโครงสร้างแฟ้มข้อมูลจุฬาจารึก มีโครงสร้างข้อมูลดังนี้

```
WORD        id,            // ตัวระบุส่วน Overlay Drawing มีค่า 0x084E
DWORD       size;          // ขนาดของส่วน Overlay
DWORD       number;        // จำนวนของรูปวาด
OVERLAYINFO overlayinfo[ ]; // รายละเอียดรูปวาด (ดูเพิ่มเติม 4.9.1)
```

4.9.1 OVERLAYINFO มีโครงสร้างข้อมูลดังนี้

```
BYTE        type;          // เป็น 1 = สี่เหลี่ยม, 2 = เส้นตรง
BYTE        FillColor;     // สีของรูปวาด
BYTE        FillPattern;   // ลวดลายของรูปวาด
BYTE        PatColor;      // สีของลวดลาย
BYTE        LineStyleWidth; // แบบและความกว้างของเส้น
BYTE        LineColor;     // สีของเส้น
BYTE        LineFlag;      // แฟล็กของเส้น
BYTE        ArrowHead;     // หัวลูกศร
BYTE        HVPos;         // จุดอ้างอิงแนวนอนคือขอบเผื่อด้านซ้ายและ
                             แนวตั้งคือบนสุดของย่อหน้า
int         xPos;          // ระยะห่างแนวนอนจากจุดอ้างอิง หน่วยเป็น twip
int         yPos;          // ระยะห่างแนวตั้งจากจุดอ้างอิง หน่วยเป็น twip
int         width;         // ความกว้างของรูปวาด หน่วยเป็น twip
int         height;        // ความสูงของรูปวาด หน่วยเป็น twip
int         numpoint;      // ยังไม่พัฒนา
```

4.10 ส่วน TTF Embed มีโครงสร้างข้อมูลดังนี้

```
WORD      id;       // ตัวระบุส่วน TTF Embed มีค่า 0x094E
DWORD     size;     // ขนาดของส่วน TTF Embed
DWORD     number;   // จำนวน TTF Embed
```

ขั้นตอนวิธีการแปลงเป็นโครงสร้างแฟ้มข้อมูลจุฬาจารึก 78

จะเห็นว่าแฟ้มข้อมูลข้อความ แฟ้มข้อมูลเวิร์ดจุฬา และแฟ้มข้อมูลจุฬาจารึก 77 จะมีโครงสร้างแฟ้มข้อมูลคล้ายกัน กล่าวคือข้อมูลจะเก็บเรียงลำดับทีละตัวอักษรตามรหัสแอสกีไปเรื่อยๆ แต่มีข้อแตกต่างกันไปบ้างเช่น มีรหัสอื่นแทรกอยู่ มีรูปภาพรวมปนอยู่ จัดเก็บตัวเลือกไว้ท้ายแฟ้มข้อมูล หรืออื่นๆ ในขณะที่จุฬาจารึก 78 มีโครงสร้างแฟ้มข้อมูลต่างออกไป โดยจัดเก็บข้อมูลแต่ละกลุ่มแยกส่วนกันไปเช่น รูปภาพจัดเก็บในส่วน Picture การจัดหน้ากระดาษเก็บในส่วน Section ข้อความจัดเก็บในส่วน Text ลักษณะประจำของข้อความจัดเก็บในส่วน Attributeรูปวาดจัดเก็บในส่วน Overlay Drawing เป็นต้น ขั้นตอนวิธีการแปลงเป็นโครงสร้างแฟ้มข้อมูลจุฬาจารึก 78 จะอ่านข้อมูลทีละรหัสจากแฟ้มข้อมูลเดิมแล้วแยกจัดเก็บลงแต่ละส่วนของโครงสร้างแฟ้มข้อมูลจุฬาจารึก 78 ในที่นี้จะกล่าวถึงขั้นตอนวิธีการแปลงข้อมูลทีละรหัสเพื่อจัดเก็บลงแต่ละส่วนของโครงสร้างแฟ้มข้อมูลจุฬาจารึก 78

1. รหัสขึ้นบรรทัดใหม่

รหัสขึ้นบรรทัดใหม่มีขนาด 2 ไบต์คือ 0x0D 0x0A โปรแกรมประมวลผลคำทั่วไปจะมีคุณสมบัติการตัดคำ เมื่อพิมพ์ข้อความจนถึงขอบเผื่อด้านขวาโปรแกรมจะตัดคำแล้วขึ้นบรรทัดใหม่ให้โดยอัตโนมัติ รหัสขึ้นบรรทัดใหม่จะเกิดจากการกดแป้น Enter ผู้ใช้ควรกดแป้นนี้เมื่อจบย่อหน้าเท่านั้นเพื่อให้สามารถจัดรูปแบบเอกสารได้สะดวก รหัสขึ้นบรรทัดใหม่มีผลต่อการแปลงโครงสร้างแฟ้มข้อมูลจุฬาจารึก 4 ส่วนคือ ส่วน Text รหัสนี้จะเก็บเป็นรหัสจบย่อหน้า (0x0D) ในแถวลำดับของข้อความ ส่วน Section แต่ละรหัสจบย่อหน้าที่เพิ่มในแถวลำดับข้อความจะเพิ่มจำนวนย่อหน้าเนื้อเรื่องด้วย ส่วน Index และส่วน Attribute รหัสนี้จะเพิ่มจำนวนตัวอักษรของข้อความที่ใช้ดรรชนีหรือลักษณะประจำ

2. <u>รหัสตั้งระยะและรหัสรูปภาพ</u>

รหัสตั้งระยะมีขนาด 1 ไบต์คือ 0x09 รหัสนี้จะพบในแฟ้มข้อมูลข้อความที่สร้างจากโปรแกรมประมวลผลคำทั่วไป เมื่อกดแป้น Tab ในโปรแกรมประมวลผลคำเคอร์เซอร์จะกระโดดไปยังจุดที่ตั้งระยะไว้โดยจัดเก็บรหัส 0x09 ไว้ในแฟ้มข้อมูล แต่เวิร์ดจุฬาจะเติมช่องว่าง (0x20) แทน รหัสตั้งระยะมีผลต่อการแปลงโครงสร้างแฟ้มข้อมูลจุฬาจารึกส่วน Text รหัสนี้จะเก็บเป็นรหัส 0x09 ในแถวลำดับของข้อความ ส่วน Attribute และส่วน Index รหัสนี้จะเพิ่มจำนวนตัวอักษรของข้อความที่ใช้ดรรชนีหรือลักษณะประจำ

นอกจากนี้รหัส 0x09 ยังพบในแฟ้มข้อมูลจุฬาจารึก 77 ซึ่งเป็นรหัสรูปภาพที่มีขนาด 3 ไบต์คือ 0x09 0x0D 0x0A แล้วตามด้วยโครงสร้างข้อมูล picnode ขนาดของรูปภาพและข้อมูลรูปภาพ รหัสรูปภาพมีผลต่อการแปลงโครงสร้างแฟ้มข้อมูลจุฬาจารึก 4 ส่วนคือส่วน Text จะเก็บรหัส 0x10 ในแถวลำดับของข้อความ ส่วน Picture จะจัดเก็บข้อมูลรูปภาพ ส่วน Index จะจัดเก็บดรรชนีรูปภาพ และส่วน Attribute จะเพิ่มจำนวนตัวอักษรที่ใช้ลักษณะประจำ

จุฬาจารึก 77 ใช้ตัวอักษรที่มีความกว้างคงที่ จึงจัดเก็บความกว้างของรูปภาพเป็นคอลัมน์ในหน่วยของหนึ่งร้อย (dx) และความยาวของรูปภาพเป็นบรรทัดในหน่วยของหนึ่งร้อย (dy) รูปภาพสามารถเคลื่อนย้ายได้ในแนวนอนเท่านั้นโดยการเปลี่ยนค่าของจุดเริ่มต้นคอลัมน์ของรูปภาพ (x) สามารถย่อ-ขยายคงสัดส่วนเดิมโดยเปลี่ยนค่าร้อยละของการย่อ-ขยายที่เท่ากัน (mx,my) หรือย่อ-ขยายสัดส่วนไม่เท่าเดิมโดยเปลี่ยนค่าร้อยละของการย่อ-ขยายที่ไม่เท่ากัน ส่วนจุฬาจารึก 78 จัดเก็บความกว้างและความยาวของรูปภาพในหน่วย twip (โดย 1 นิ้วเท่ากับ 1440 twip) ความกว้างของรูปภาพในหน่วยหนึ่งร้อยของคอลัมน์สามารถแปลงเป็นหน่วย twip ได้โดยใช้ค่าตัวอักษรต่อนิ้วที่จัดเก็บในตัวเลือกท้ายแฟ้มข้อมูล ส่วนความสูงของรูปภาพไม่สามารถหาได้โดยตรงจากจำนวนบรรทัด สามารถหาได้จากสัดส่วนความสูงเดิมต่อความกว้างเดิมของรูปภาพและค่าร้อยละของการย่อ-ขยาย รูปภาพแบบ WMF จัดเก็บความกว้างเดิม (xExt) และความสูงเดิม (yExt) ใน mfpic ของโครงสร้างข้อมูล picnode รูปภาพแบบ BMP จัดเก็บความกว้างเดิม (biWidth) และความสูงเดิม (biHeight) ในโครงสร้างข้อมูล bmiHeader สมการที่ใช้ในการแปลงคือ

$$\text{width} = \frac{dx * 1440}{100 * \text{จำนวนตัวอักษรต่อนิ้ว}}$$

$$orgwidth = \frac{width * 100}{mx}$$

$$height \text{ ของ } WMF = \frac{width * yExt * my}{xExt * mx}$$

$$height \text{ ของ } BMP = \frac{width * biHeight * my}{biWidth * mx}$$

$$orgheight = \frac{height * 100}{my}$$

$$vertadj = \frac{x * 1440}{\text{จำนวนตัวอักษรต่อนิ้ว}}$$

โดย

width	เป็นความกว้างของรูปภาพในหน่วย twip
height	เป็นความสูงของรูปภาพในหน่วย twip
orgwidth	เป็นความกว้างเดิมของรูปภาพในหน่วย twip
orgheight	เป็นความสูงเดิมของรูปภาพในหน่วย twip
vertadj	เป็นจุดเริ่มต้นของรูปแบบในหน่วย twip
dx	เป็นความกว้างของรูปภาพในหน่วยหนึ่งร้อยของคอลัมน์
dy	เป็นความสูงของรูปภาพในหน่วยหนึ่งร้อยของบรรทัด
mx	เป็นค่าร้อยละของการย่อ-ขยายในแนวแกน x
my	เป็นค่าร้อยละของการย่อ-ขยายในแนวแกน y
xExt	เป็นความกว้างเดิมของรูปภาพแบบ WMF
yExt	เป็นความสูงเดิมของรูปภาพแบบ WMF
biWidth	เป็นความกว้างเดิมของรูปภาพแบบ BMP
biHeight	เป็นความสูงเดิมของรูปภาพแบบ BMP

3. <u>รหัสตัดคำ</u>

รหัสตัดคำมีขนาด 3 ไบต์คือ 0x8D 0x0D 0x0A เวิร์ดจุฬาและจุฬาจารึกมีคุณสมบัติการตัดคำ เมื่อพิมพ์ข้อความจนถึงขอบเผื่อด้านขวาโปรแกรมจะตัดคำแล้วขึ้นบรรทัด

ใหม่ให้โดยอัตโนมัติ อย่างไรก็ตามรหัสตัดคำนี้ไม่มีผลต่อการแปลงโครงสร้างแฟ้มข้อมูลจุฬาจารึกเพราะจุฬาจารึกจะจัดการตัดคำเอง

4. รหัสรูปแบบพิเศษ

รหัสรูปแบบพิเศษแสดงดังตารางที่ 2.1 มีผลต่อการแปลงโครงสร้างแฟ้มข้อมูลจุฬาจารึก 3 ส่วนคือ ส่วน Font ส่วน Index และส่วน Attribute จะเห็นว่าแบบอักษรจะจัดเก็บชื่อแบบอักษร ขนาดตัวอักษร รูปแบบตัวหนา/ตัวเอียง ระยะห่างของตัวยก/ตัวห้อยจากบรรทัด ดังนั้นรูปแบบพิเศษตัวหนา ตัวเอียง ตัวใหญ่ ตัวยกและตัวห้อย จึงมีผลต่อส่วน Font ซึ่งจัดเก็บแบบอักษรที่ใช้ในเอกสาร และส่วน Index ซึ่งจัดเก็บดรรชนีแบบอักษรที่ข้อความใช้และจำนวนตัวอักษรของข้อความที่รหัสรูปแบบพิเศษดังกล่าวปิดกำกับหัวท้าย นอกจากนี้บิตที่ 4-7 ของลักษณะประจำจัดเก็บลักษณะตัวยก ตัวห้อย ตัวขีดเส้นใต้ 1 เส้นและตัวขีดเส้นใต้ 2 เส้น ดังนั้นรูปแบบพิเศษตัวยก ตัวห้อย ตัวขีดเส้นใต้ 1 เส้นและตัวขีดเส้นใต้ 2 เส้น จึงมีผลต่อส่วน Attribute ซึ่งจัดเก็บลักษณะประจำที่ข้อความใช้และจำนวนตัวอักษรของข้อความที่รหัสรูปแบบพิเศษดังกล่าวปิดกำกับหัวท้าย

5. รหัสตาราง

รหัสตารางแสดงดังตารางที่ 2.2 เวิร์ดจุฬาได้เตรียมตัวอักษรกราฟฟิกของเส้นส่วนประกอบตาราง จึงจัดเก็บตารางทีละอักษรเช่นเดียวกับตัวอักษรของข้อความ แต่จุฬาจารึกไม่ได้เตรียมตัวอักษรกราฟฟิกของเส้นส่วนประกอบตาราง จึงสร้างตารางโดยวาดรูปเส้นตรงแทน ดังนั้นรหัสตารางจึงมีผลต่อการแปลงโครงสร้างแฟ้มข้อมูลจุฬาจารึกส่วน Overlay Drawing ซึ่งจัดเก็บข้อมูลรูปวาดเส้นตรง และส่วน Section ซึ่งจัดเก็บจำนวนรูปวาดที่มีในแต่ละย่อหน้า

เวิร์ดจุฬาและจุฬาจารึก 77 ใช้ตัวอักษรที่มีความกว้างเท่ากันหมดทุกตัวซึ่งทำให้การแสดงผลไม่สวยงาม เช่นตัว า มีความกว้างจริงน้อยกว่าตัว ญ เมื่อนำมาจัดวางให้ความกว้างเท่ากัน ตัวอักษรที่มีความกว้างน้อยจะอยู่ห่างกันเกิดเป็นช่องว่าง ต่อมาจุฬาจารึก 78 ได้ใช้ตัวอักษรที่มีความกว้างไม่คงที่ซึ่งทำให้การแสดงผลสวยงาม ตัวอักษรที่มีความกว้างน้อยจะไม่อยู่ห่างกันจึงไม่เกิดเป็นช่องว่างขึ้น แต่เกิดปัญหาในการแปลงเป็นโครงสร้างแฟ้มข้อมูลจุฬาจารึก 78 ข้อความที่มีตัวอักษรความกว้างมากจะยาวกว่าข้อความที่มีตัวอักษรความกว้างน้อย ทำให้ความยาวของข้อความในตารางแต่ละคอลัมน์ไม่คงที่ จำเป็นต้องเติมช่องว่างให้ข้อความในตารางจนมี

ความยาวใกล้เคียงกัน นอกจากนี้เวิร์ดจุฬาและจุฬาจารึก 77 สามารถใช้ตัวอักษรได้เพียงขนาดเดียวตลอดเอกสาร แต่ละบรรทัดจึงมีความสูงเท่ากันตลอด แต่จุฬาจารึก 78 สามารถใช้ตัวอักษรได้หลายขนาดในเอกสารเดียวกัน แต่ละบรรทัดจึงมีความสูงไม่คงที่ขึ้นอยู่กับขนาดตัวอักษร ดังนั้นบรรทัดที่มีตัวอักษรใหญ่จึงสูงกว่าบรรทัดที่มีตัวอักษรปกติ ดังนั้นการแปลงเป็นโครงสร้างแฟ้มข้อมูลจุฬาจารึก 78 จำเป็นต้องหาความสูงของแต่ละบรรทัดด้วย จุฬาจารึก 78 จะใช้รูปวาดเส้นตรงสร้างตารางโดยค้นหาตำแหน่งคอลัมน์และตำแหน่งบรรทัดของรหัสตารางแล้วคำนวณหาตำแหน่งแกน x ตำแหน่งแกน y ความกว้างและความสูงของแต่ละเส้นตรงของตาราง

6. รหัสแทรกจากการตัดคำ

รหัสแทรกจากการตัดคำมีขนาด 1 ไบต์คือ 0xA0 เมื่อพิมพ์ข้อความในเวิร์ดจุฬารุ่นที่ 1.52 จนถึงขอบเผื่อด้านขวาโปรแกรมจะตัดคำโดยแทรกช่องว่าง (0x20) จนเต็มบรรทัดก่อนขึ้นบรรทัดใหม่ให้โดยอัตโนมัติ นอกจากนี้ช่องว่างยังเกิดจากการกดแป้น Spacebar เพื่อเว้นวรรคข้อความด้วย ในการแปลงโครงสร้างแฟ้มข้อมูลจึงไม่สามารถตัดช่องว่างที่แทรกจากการตัดคำได้ แต่เวิร์ดจุฬารุ่นที่ 1.6 และจุฬาจารึก 77 ได้เปลี่ยนรหัสแทรกจากการตัดคำเป็น 0xA0 แทนช่องว่าง ในการแปลงโครงสร้างแฟ้มข้อมูลจึงตัดรหัสแทรกที่ไม่จำเป็นทิ้งได้

7. รหัสแบบอักษร

รหัสแบบอักษรแสดงดังตารางที่ 2.3 เวิร์ดจุฬารุ่นที่ 1.6 ได้เตรียมแบบอักษรไว้ให้ใช้ได้ 4 แบบ แต่จุฬาจารึกสามารถเลือกใช้แบบอักษรทรูไทป์ของวินโดวได้ ดังนั้นรหัสนี้จึงไม่มีผลต่อการแปลงโครงสร้างแฟ้มข้อมูลเป็นจุฬาจารึก

8. สมการคณิตศาสตร์

สมการคณิตศาสตร์แสดงดังตารางที่ 2.4 เวิร์ดจุฬารุ่นที่ 1.6 ได้ปรับปรุงให้สร้างสมการคณิตศาสตร์ได้ ถ้าบรรทัดใดมีคำสั่งจุด M ที่คอลัมน์ 1 จะสร้างสมการคณิตศาสตร์ตามฟังก์ชันนั้น แต่จุฬาจารึกไม่สามารถสร้างสมการคณิตศาสตร์ได้ ดังนั้นสมการคณิตศาสตร์จะแปลงเป็นข้อความเท่านั้น จึงต้องตัดคำสั่งจุด M เครื่องหมายคำพูดคร่อมข้อความ และตัวเชื่อมข้อความ (~) ทิ้ง จึงมีผลต่อการแปลงโครงสร้างแฟ้มข้อมูลจุฬาจารึกส่วน Text จะเก็บข้อความใน

แถวลำดับของข้อความ ส่วน Index และส่วน Attribute จะเพิ่มจำนวนตัวอักษรของข้อความที่ใช้ดรรชนีหรือลักษณะประจำ

9. ตัวอักษรตามรหัสแอสกี

รหัสแอสกีจำนวน 256 ตัวจะเป็นอักษรภาษาอังกฤษตั้งแต่รหัส 0x20 จนถึงรหัส 0x7E เป็นอักษรภาษาไทยตั้งแต่รหัส 0xA1ขึ้นไป เวิร์ดจุฬาและจุฬาจารึก 77 สามารถใช้รหัสภาษาไทย 2 รหัสคือรหัสสมอ. และรหัสเกษตร แต่จุฬาจารึก 78 สามารถใช้รหัสสมอ. เท่านั้น ถ้าแฟ้มข้อมูลเดิมใช้รหัสเกษตรจำเป็นต้องแปลงเป็นรหัสสมอ. ก่อนแปลงโครงสร้างแฟ้มข้อมูลโดยใช้ตาราง kutostd ซึ่งเป็นตารางที่ประกอบด้วยรหัสภาษาไทยจำนวน128 ตัว จะนำรหัสภาษาไทยรหัสเกษตรลบด้วยค่า 128 แล้วนำผลลบไปค้นหาข้อมูลในลำดับที่นั้นของตาราง ตัวอักษรตามรหัสแอสกีจึงมีผลต่อการแปลงโครงสร้างแฟ้มข้อมูลจุฬาจารึกส่วน Text จะเก็บข้อความในแถวลำดับของข้อความ ส่วน Index และส่วน Attribute จะเพิ่มจำนวนตัวอักษรของข้อความที่ใช้ดรรชนีหรือลักษณะประจำ

การออกแบบและพัฒนาโปรแกรมอรรถประโยชน์ด้านการแปลงโครงสร้างแฟ้มข้อมูล

โครงสร้างแฟ้มข้อมูลจุฬาจารึก 78 ได้จัดเก็บข้อมูลแต่ละกลุ่มแยกเป็นส่วนต่างๆ 10 ส่วน เมื่อต้องการแปลงโครงสร้างแฟ้มข้อมูลจะถูกอ่านข้อมูลจากแฟ้มข้อมูลเดิมเข้าทีละตัวแล้วแยกจัดเก็บลงในแต่ละส่วน ขั้นตอนวิธีการแปลงเป็นโครงสร้างแฟ้มข้อมูลจุฬาจารึก 78 กล่าวไว้ในหัวข้อที่แล้วและแสดงดังรูปที่ 2.10 วิธีการแปลงเป็นโครงสร้างแฟ้มข้อมูลจุฬาจารึกมี 4 ขั้นตอนคือ

ก. การกำหนดค่าเริ่มต้นต่างๆที่จำเป็นของแต่ละส่วนเช่น การจัดหน้ากระดาษ ขนาดตัวอักษร การจัดย่อหน้า ขอบเผื่อด้านบน/ล่าง/ซ้าย/ขวา เป็นต้น

ข. ตรวจสอบค่าตัวเลือกท้ายแฟ้มข้อมูล ตรวจสอบเครื่องพิมพ์ที่ระบบติดตั้ง ตรวจสอบแบบอักษรภาษาไทยที่มีในระบบ และสร้างแบบอักษรภาษาไทยสำหรับใช้ในโปรแกรม

ค. อ่านข้อมูลจากแฟ้มข้อมูลเดิม แล้วแปลงข้อมูลแต่ละตัวแยกเป็นส่วนต่างๆ (รายละเอียดการแปลงข้อมูลแต่ละตัวได้กล่าวไว้ในบทที่ 2)

ง. นำข้อมูลที่แยกเป็นแต่ละส่วน มาจัดเก็บลงตามโครงสร้างแฟ้มข้อมูลจุฬาจารึก

1. โครงสร้างข้อมูลที่ใช้ในโปรแกรม

โครงสร้างข้อมูลที่ใช้ในโปรแกรมอรรถประโยชน์ด้านการแปลงโครงสร้างแฟ้มข้อมูลที่สำคัญดังนี้

1.1 ค่าคงที่ เป็นค่าที่กำหนดไว้ตั้งแต่ต้นโปรแกรมโดยไม่มีการเปลี่ยนแปลง ค่าคงที่แสดงดังตารางที่ 2.5

ชื่อค่าคงที่	ค่าที่เก็บ	คำอธิบาย
ENDCW	0x1A	รหัสจบแฟ้มข้อมูลเวิร์ดจุฬา
CR	0x0D	รหัส CR
LF	0x0A	รหัส LF
TAB	0x09	รหัส TAB
WRAP	0x8D	รหัสตัดคำ
ITALIC	0x17	รหัสรูปแบบตัวเอียง
UNDERLINE	0x13	รหัสรูปแบบตัวขีดเส้นใต้ 1 เส้น
DOUBLELINE	0x12	รหัสรูปแบบตัวขีดเส้นใต้ 2 เส้น
BOLD	0x02	รหัสรูปแบบตัวหนา
ENLARGE	0x05	รหัสรูปแบบตัวใหญ่
SUPERSCRIPT	0x14	รหัสรูปแบบตัวยก
SUBSCRIPT	0x16	รหัสรูปแบบตัวห้อย
UL	0x98	รหัสตารางขอบบนด้านซ้าย
UM	0x91	รหัสตารางขอบบนตรงกลาง
UR	0x99	รหัสตารางขอบบนด้านขวา
ML	0x93	รหัสตารางกลางด้านซ้าย
MM	0x8F	รหัสตารางกึ่งกลาง
MR	0x92	รหัสตารางกลางด้านขวา
BL	0x9A	รหัสตารางขอบล่างด้านซ้าย

ตารางที่ 2.5 แสดงค่าคงที่ในโปรแกรมอรรถประโยชน์ด้านการแปลงโครงสร้างแฟ้มข้อมูล

ชื่อค่าคงที่	ค่าที่เก็บ	คำอธิบาย
BM	0x90	รหัสตารางขอบล่างตรงกลาง
BR	0x9B	รหัสตารางขอบล่างด้านขวา
V	0x96	รหัสตารางเส้นตั้ง
H	0x95	รหัสตารางเส้นนอน
FONT1	0x1B	รหัสแบบอักษร 1
FONT2	0x1C	รหัสแบบอักษร 2
FONT3	0x1D	รหัสแบบอักษร 3
FONT4	0x1E	รหัสแบบอักษร 4
WRAP77	0xA0	รหัสแทรกจากการตัดคำแทนช่องว่าง
DOT	0x2E	ตัวอักษรจุด (.)
CONNECT	0x7E	ตัวอักษร ~
QUOTATION	0x22	ตัวอักษรเครื่องหมายคำพูด
M	0x4D	ตัวอักษร M
NUMFONT	12	จำนวนแบบอักษรที่เป็นไปได้
STYLENAME	"Normal"	ชื่อสไตล์
FONTNAME	"AngsanaUPC"	ชื่อแบบอักษร
NormalPt	16	ขนาดตัวอักษรตัวปกติ หน่วยเป็น point
EnlargePt	20	ขนาดตัวอักษรตัวใหญ่ หน่วยเป็น point
TWIP	1440	1 นิ้วเท่ากับ 1440 twip
OPTLENGTH	1232	ขนาดสูงสุดของตัวเลือกท้ายแฟ้มข้อมูล
CPI	12	จำนวนตัวอักษรต่อนิ้ว
PIC	0x10	รหัสรูปภาพในแถวลำดับข้อความของส่วน Text
BK	0x20	รหัสช่องว่างในแถวลำดับข้อความของส่วน Text
WMF	1	ค่ารหัสรูปภาพแบบ WMF ของส่วน Picture
BMP	2	ค่ารหัสรูปภาพแบบ BMP ของส่วน Picture
LINE	2	ค่ารหัสรูปวาดเส้นตรงของส่วน Overlay Drawing

ตารางที่ 2.5 (ต่อ) แสดงค่าคงที่ในโปรแกรมอรรถประโยชน์ด้านการแปลงโครงสร้างแฟ้มข้อมูล

1.2 ตัวแปร จะกล่าวเฉพาะตัวแปรที่สำคัญที่ใช้ในโปรแกรมอรรถประโยชน์ด้านการแปลงโครงสร้างแฟ้มข้อมูล ดังนี้

header	เป็นตัวแปรโครงสร้างของส่วน Header (โครงสร้างข้อมูลกล่าวไว้ในหัวข้อที่แล้ว)
style	เป็นตัวแปรโครงสร้างของส่วน Style (โครงสร้างข้อมูลกล่าวไว้ในหัวข้อที่แล้ว)
section	เป็นตัวแปรโครงสร้างของส่วน Section (โครงสร้างข้อมูลกล่าวไว้ในหัวข้อที่แล้ว)
numbodypara	เป็นตัวแปรชนิดตัวเลข จัดเก็บจำนวนย่อหน้า
paragraphinfo	เป็นตัวแปรโครงสร้างของย่อหน้า (โครงสร้างข้อมูลกล่าวไว้ในหัวข้อที่แล้ว)
font	เป็นตัวแปรโครงสร้างของส่วน Font (โครงสร้างข้อมูลกล่าวไว้ในหัวข้อที่แล้ว)
fontinfo	เป็นตัวแปรโครงสร้างของรายละเอียดแบบอักษร (โครงสร้างข้อมูลกล่าวไว้ในหัวข้อที่แล้ว)
picture	เป็นตัวแปรโครงสร้างของส่วน Picture (โครงสร้างข้อมูลกล่าวไว้ในหัวข้อที่แล้ว)
pictureinfo	เป็นตัวแปรโครงสร้างของรายละเอียดรูปภาพ (โครงสร้างข้อมูลในหัวข้อที่แล้ว)
text	เป็นตัวแปรโครงสร้างของส่วน Text (โครงสร้างข้อมูลกล่าวไว้ในหัวข้อที่แล้ว)
index	เป็นตัวแปรโครงสร้างของส่วน Index (โครงสร้างข้อมูลกล่าวไว้ในหัวข้อที่แล้ว)
indexinfo	เป็นตัวแปรโครงสร้างของรายละเอียดดรรชนี (โครงสร้างข้อมูลในหัวข้อที่แล้ว)
attribute	เป็นตัวแปรโครงสร้างของส่วน Attribute (โครงสร้างข้อมูลกล่าวในหัวข้อที่แล้ว)
attributeinfo	เป็นตัวแปรโครงสร้างของรายละเอียดลักษณะประจำ (โครงสร้างข้อมูลกล่าวไว้ในหัวข้อที่แล้ว)
overlay	เป็นตัวแปรโครงสร้างของส่วน Overlay Drawing (โครงสร้างข้อมูลกล่าวไว้ในหัวข้อที่แล้ว)
overlayinfo	เป็นตัวแปรโครงสร้างของรายละเอียดรูปภาพ (โครงสร้างข้อมูลกล่าวไว้ในหัวข้อที่แล้ว)
ttfembed	เป็นตัวแปรโครงสร้างของส่วน TTFEmbed (โครงสร้างข้อมูลในหัวข้อที่แล้ว)
fsize	เป็นตัวแปรชนิดตัวเลข จัดเก็บขนาดแฟ้มข้อมูลที่ต้องการแปลงโครงสร้าง
size	เป็นตัวแปรชนิดตัวเลข จัดเก็บขนาดข้อมูลที่อ่านได้จากแฟ้มข้อมูล
buffer	เป็นตัวแปรแถวลำดับชนิดตัวอักษรขนาด10000 ใช้สำหรับอ่านข้อความจากแฟ้มข้อมูลและจัดเก็บข้อความของส่วน Text
line	เป็นตัวแปรแถวลำดับชนิดตัวอักษรขนาด 256 ใช้สำหรับหาความยาวข้อความ
notend	เป็นตัวแปรชนิดตัวเลข ใช้ตรวจว่าพบรหัสจบแฟ้มข้อมูลเวิร์ดจุฬาหรือไม่

ถ้าเป็น 0 แสดงว่า ไม่พบรหัสจบแฟ้มข้อมูลเวิร์ดจุฬา

ถ้าเป็น -1 แสดงว่า พบรหัสจบแฟ้มข้อมูลเวิร์ดจุฬา

math เป็นตัวแปรชนิดตัวเลข ใช้ตรวจว่าสมการคณิตศาสตร์

ถ้าเป็น -1 แสดงว่า เป็นสมการคณิตศาสตร์

ถ้าไม่เป็น -1 แสดงว่า ไม่เป็นสมการคณิตศาสตร์

kaset เป็นตัวแปรชนิดตัวเลข ใช้บอกสภาวะการใช้รหัสภาษาไทย

ถ้าเป็น 0 แสดงว่า ใช้รหัสเกษตร

ถ้าเป็น 1 แสดงว่า ใช้รหัสสมอ.

table เป็นตัวแปรชนิดตัวเลข ใช้บอกสภาวะการสร้างตาราง

ถ้าเป็น 0 แสดงว่า ไม่ได้สร้างตารางอยู่

ถ้ามากกว่า 0 แสดงว่า สร้างตารางอยู่

c เป็นตัวแปรชนิดตัวเลข จัดเก็บคอลัมน์ของตาราง

r เป็นตัวแปรชนิดตัวเลข จัดเก็บแถวของตาราง

col เป็นตัวแปรแถวลำดับชนิดตัวเลขขนาด 256 ใช้จัดเก็บความกว้างคอลัมน์

row เป็นตัวแปรแถวลำดับชนิดตัวเลขขนาด 80 ใช้จัดเก็บความสูงบรรทัดในตาราง

hor เป็นตัวแปรโครงสร้าง ใช้จัดเก็บคอลัมน์และแถวของเส้นตรงแนวนอน ประกอบด้วยข้อมูลดังนี้

y เป็นตัวแปรตัวเลขจัดเก็บแถวของเส้นตรงแนวนอน

x1 เป็นตัวแปรตัวเลขจัดเก็บคอลัมน์ของจุดเริ่มต้นของเส้นตรงแนวนอน

x2 เป็นตัวแปรตัวเลขจัดเก็บคอลัมน์ของจุดสุดท้ายของเส้นตรงแนวนอน

ver เป็นตัวแปรโครงสร้าง ใช้จัดเก็บคอลัมน์และแถวของเส้นตรงแนวตั้ง ประกอบด้วยข้อมูลดังนี้

x เป็นตัวแปรตัวเลขจัดเก็บคอลัมน์ของเส้นตรงแนวตั้ง

y1 เป็นตัวแปรตัวเลขจัดเก็บแถวของจุดเริ่มต้นของเส้นตรงแนวตั้ง

y2 เป็นตัวแปรดัวเลขจัดเก็บแถวของจุดสุดท้ายของเส้นตรงแนวตั้ง

cpi เป็นตัวแปรชนิดตัวเลข จัดเก็บจำนวนตัวอักษรต่อนิ้ว

picnode เป็นตัวแปรโครงสร้าง picnode ที่เก็บรายละเอียดของรูปภาพ (โครงสร้างข้อมูลกล่าวไว้ในหัวข้อที่แล้ว)

picsize	เป็นตัวแปรชนิดตัวเลขที่ใช้จัดเก็บขนาดรูปภาพ
bmiheader	เป็นตัวแปรโครงสร้าง bmiheader ที่เก็บรายละเอียดรูปภาพแบบ BMP (โครงสร้างข้อมูลกล่าวไว้ในหัวข้อที่แล้ว)
picsize	เป็นตัวแปรชนิดตัวเลขที่ใช้จัดเก็บขนาดรูปภาพ
tmpindex	เป็นแฟ้มข้อมูลชั่วคราวของส่วน Index
tmpattribute	เป็นแฟ้มข้อมูลชั่วคราวของส่วน Attribute
tmptext	เป็นแฟ้มข้อมูลชั่วคราวของส่วน Text
tmppic	เป็นแฟ้มข้อมูลชั่วคราวของส่วน Picture
tmpovr	เป็นแฟ้มข้อมูลชั่วคราวของส่วน Overlay Drawing

2. การพัฒนาโปรแกรมอรรถประโยชน์ด้านการแปลงโครงสร้างแฟ้มข้อมูล

ในที่นี้จะกล่าวถึงขั้นตอนการทำงานทั้งหมดของโปรแกรมอรรถประโยชน์ด้านการแปลงโครงสร้างแฟ้มข้อมูล ซึ่งบางขั้นตอนมีรายละเอียดมาก จำเป็นต้องใช้ผังงานหลักอธิบายขั้นตอนการทำงานทั้งหมดก่อน จากนั้นใช้ผังงานย่อยอธิบายรายละเอียดต่อไป

2.1 โปรแกรมอรรถประโยชน์ด้านการแปลงโครงสร้างแฟ้มข้อมูล ผังงานของขั้นตอนการทำงานทั้งหมดแสดงดังรูปที่ 2.10 มีขั้นตอนการทำงานดังนี้

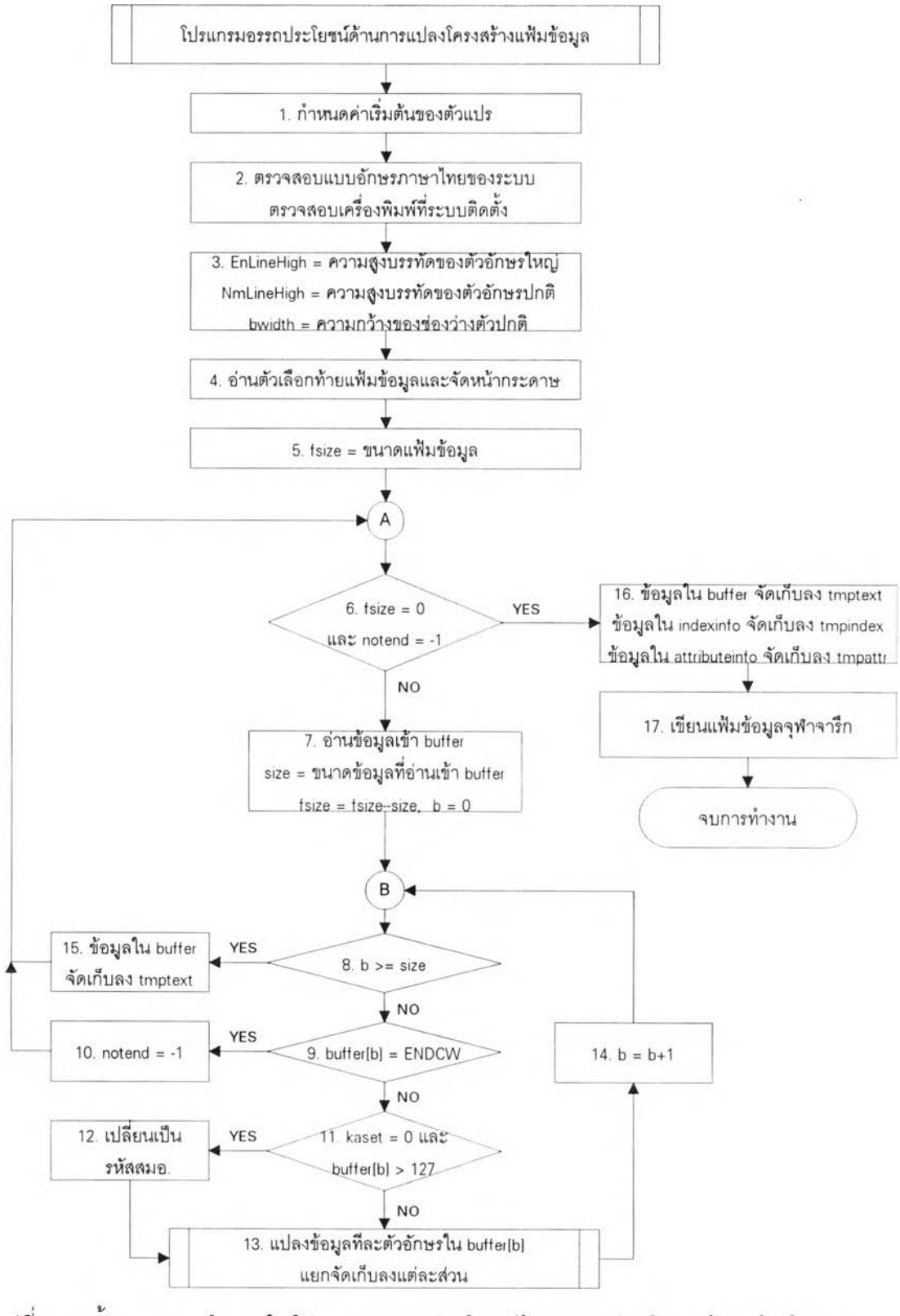
1. กำหนดค่าเริ่มต้นของตัวแปรที่ใช้ในโปรแกรม และค่าเริ่มต้นต่างๆที่จำเป็นของแต่ละส่วน

2. ตรวจสอบแบบอักษรในระบบที่มีรหัสภาษาไทยเพื่อกำหนดให้โปรแกรมใช้แบบอักษรดังกล่าว และตรวจสอบชนิดเครื่องพิมพ์ที่ระบบติดตั้ง

3. หาค่า NmLineHigh, EnLineHigh และ bwidth ดังนี้

3.1 สร้างชุดแบบอักษรตัวใหญ่ (ขนาด 20 point) ที่แสดงบนเครื่องพิมพ์และหาความสูงตัวอักษรในหน่วย twip เพื่อกำหนดให้ EnLineHigh ซึ่งเป็นความสูงบรรทัดของตัวอักษรตัวใหญ่ สำหรับใช้ในการสร้างรูปวาดเส้นตรงของตาราง

3.2 สร้างชุดแบบอักษรตัวปกติ (ขนาด 16 point) ที่แสดงบนเครื่องพิมพ์และหาความสูงตัวอักษรในหน่วย twip เพื่อกำหนดให้ NmLineHigh ซึ่งเป็นความสูงบรรทัดของตัวอักษรตัวปกติ สำหรับใช้ในการสร้างรูปวาดเส้นตรงของตาราง และหาค่าความกว้างของตัว



รูปที่ 2.10 ขั้นตอนการทำงานในโปรแกรมอรรถประโยชน์ด้านการแปลงโครงสร้างแฟ้มข้อมูล

I17522614

อักษรช่องว่างให้ bwidth สำหรับใช้ในการแปลงข้อความในตาราง

4. อ่านตัวเลือกท้ายแฟ้มข้อมูลดังนี้

4.1 อ่านค่าตัวอักษรต่อนิ้วจากตัวเลือกท้ายแฟ้มข้อมูลให้ตัวแปร cpi เพื่อใช้ในการคำนวณเป็นหน่วย twip ถ้าแฟ้มข้อมูลไม่จัดเก็บตัวเลือกท้ายแฟ้มข้อมูลจะกำหนดให้ตัวอักษรต่อนิ้วเป็น 12

4.2 อ่านค่ากั้นหน้าซ้าย, กั้นหน้าขวา, จำนวนบรรทัดต่อหน้าจากตัวเลือกท้ายแฟ้มข้อมูลเพื่อกำหนดชนิดกระดาษและการวางหน้ากระดาษ ถ้าแฟ้มข้อมูลไม่จัดเก็บตัวเลือกท้ายแฟ้มข้อมูลจะกำหนดให้ชนิดกระดาษเป็น A4 และวางหน้ากระดาษแนวตั้ง

4.3 อ่านรหัสภาษาไทยของเอกสารให้ตัวแปร kaset ถ้าแฟ้มข้อมูลไม่จัดเก็บตัวเลือกท้ายแฟ้มข้อมูลจะกำหนดให้รหัสภาษาไทยเป็นรหัสสมอ

5. หาค่า fsize ซึ่งเป็นขนาดแฟ้มข้อมูลเดิมที่ต้องการแปลงโครงสร้าง

6. ตรวจสอบว่าขนาดแฟ้มข้อมูลเป็นศูนย์หรือไม่ หรือพบรหัสจบแฟ้มข้อมูลเวิร์ดจุฬาหรือไม่ ถ้าคำตอบคือ ใช่ แสดงว่าจบการอ่านจากแฟ้มข้อมูลเดิมเพื่อแปลงข้อมูลทีละตัว จึงข้ามไปทำงานข้อ 16 ถ้าคำตอบคือ ไม่ใช่ จะทำงานต่อไป

7. อ่านข้อมูลจากแฟ้มข้อมูลเดิมเข้าสู่ buffer พร้อมทั้งลดขนาดแฟ้มข้อมูลโดย size คือขนาดข้อมูลที่อ่านเข้าสู่ buffer ได้และ b คือลำดับที่ของตัวอักษรใน buffer ที่กำลังใช้งานอยู่ เริ่มต้นให้ b มีค่าเป็นศูนย์

8. ตรวจสอบว่า b มีค่ามากกว่าหรือเท่ากับ size หรือไม่ ถ้าคำตอบคือ ใช่ แสดงว่าตัวอักษรใน buffer หมดแล้วจึงข้ามไปทำงานข้อ 15 ถ้าคำตอบคือ ไม่ใช่ จะทำงานต่อไป

9. ตรวจสอบว่า ตัวอักษรลำดับที่ b ใน buffer เป็นรหัสจบแฟ้มข้อมูลเวิร์ดจุฬาหรือไม่ ถ้าคำตอบคือ ใช่ แสดงว่าจบการอ่านจากแฟ้มข้อมูลเดิมเพื่อแปลงข้อมูลทีละตัว ถ้าคำตอบคือ ไม่ใช่ จะทำงานต่อไปในข้อ 11

10. กำหนดให้ notend เท่ากับ -1 เพื่อแสดงว่าพบรหัสจบแฟ้มข้อมูลเวิร์ดจุฬา เมื่อย้อนกลับไปทำงานข้อ 6 จะออกจากการอ่านจากแฟ้มข้อมูลเดิมเพื่อแปลงข้อมูลทีละตัว

11. ตรวจสอบว่า kaset เท่ากับ 0 หรือไม่ และตัวอักษรลำดับที่ b ใน buffer มีค่ามากกว่า 127 หรือไม่ คำตอบคือ ใช่ แสดงว่าเป็นตัวอักษรภาษาไทยรหัสเกษตร ถ้าคำตอบคือ ไม่ใช่ แสดงว่าเป็นตัวอักษรภาษาอังกฤษ หรือเป็นตัวอักษรภาษาไทยรหัสสมอ จะทำงานต่อไปในข้อ 13

12. เปลี่ยนตัวอักษรภาษาไทยจากรหัสเกษตรเป็นรหัสสมอ โดยนำตัว

อักษรภาษาไทยรหัสเกษตรลบด้วย 128 แล้วนำผลลบไปหาข้อมูลในลำดับที่นั้นของตารางkutostd

13. แปลงตัวอักษรลำดับที่ b ใน buffer ทีละตัวเพื่อแยกจัดเก็บลงแต่ละส่วน (รายละเอียดดูเพิ่มเติมในผังงานย่อยแสดงดังรูปที่ 2.11)

14. เพิ่ม b อีก 1 เพื่อใช้งานตัวอักษรถัดไป แล้วย้อนกลับไปทำงานข้อ 8

15. ในขั้นตอนการแปลงตัวอักษรทีละตัวเพื่อแยกจัดเก็บลงแต่ละส่วน จะจัดเก็บข้อมูลส่วน Text ไว้ใน buffer ก่อน เมื่อตัวอักษรที่ต้องการแปลงใน buffer หมดแล้วจะวนกลับไปอ่านจากแฟ้มข้อมูลเดิม โดยจัดเก็บข้อมูลส่วน Text ใน buffer ลงแฟ้มข้อมูลชั่วคราว tmptext ก่อน

16. เมื่ออ่านจากแฟ้มข้อมูลเดิมหมดแล้ว หรือพบรหัสจบแฟ้มข้อมูลเวิร์ดจุฬาแสดงว่าจบขั้นตอนการแปลงข้อมูลทีละตัวแล้ว จะจัดเก็บข้อมูลส่วน Text ใน buffer ลงแฟ้มข้อมูลชั่วคราว tmptext จัดเก็บข้อมูลส่วน Index ใน indexinfo ลงแฟ้มข้อมูลชั่วคราว tmpindex และจัดเก็บข้อมูลส่วน Attribute ใน attrinfo ลงแฟ้มข้อมูลชั่วคราว tmpattr

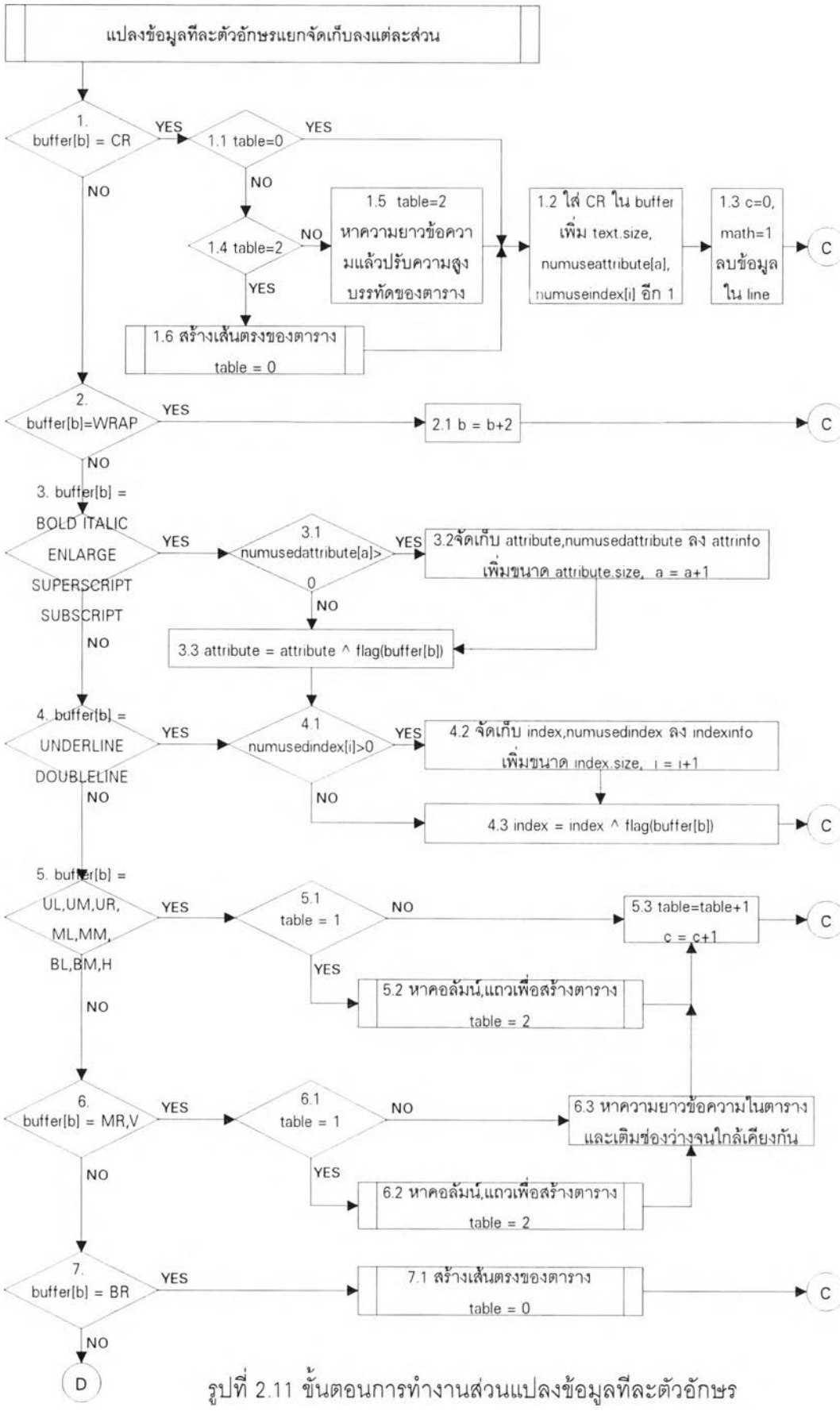
17. เขียนแฟ้มข้อมูลจุฬาจารึก เริ่มจากคำนวณจุดเริ่มต้นของแต่ละส่วนของแฟ้มข้อมูลจุฬาจารึก และจัดเก็บจุดเริ่มต้นเหล่านั้นในส่วน Header จากนั้นนำข้อมูลที่แยกเป็นแต่ละส่วนมาจัดเก็บลงตามโครงสร้างแฟ้มข้อมูลจุฬาจารึก โดยขั้นตอนการอ่านข้อมูลแล้วแปลงข้อมูลแยกเป็นแต่ละส่วนนั้น ได้จัดเก็บข้อมูลส่วน Text ในแฟ้มข้อมูลชั่วคราว tmptext ข้อมูลส่วน Index จัดเก็บลงแฟ้มข้อมูลชั่วคราว tmpindex ข้อมูลส่วน Attribute จัดเก็บลงแฟ้มข้อมูลชั่วคราว tmpattr ข้อมูลส่วน Picture จัดเก็บลงแฟ้มข้อมูลชั่วคราว tmppic ข้อมูลส่วน Overlay จัดเก็บลงแฟ้มข้อมูลชั่วคราว tmpover และข้อมูลส่วนที่เหลือจัดเก็บในตัวแปรที่กำหนด

2.2 ส่วนแปลงข้อมูลทีละตัวอักษรแยกจัดเก็บลงแต่ละส่วน ขั้นตอนนี้จะตรวจสอบตัวอักษรลำดับที่ b ใน buffer ว่าเป็นรหัสใดแล้วแยกไปทำงานแต่ละขั้นตอนต่อไป ผังงานย่อยแสดงดังรูปที่ 2.11 มีขั้นตอนการทำงานดังนี้

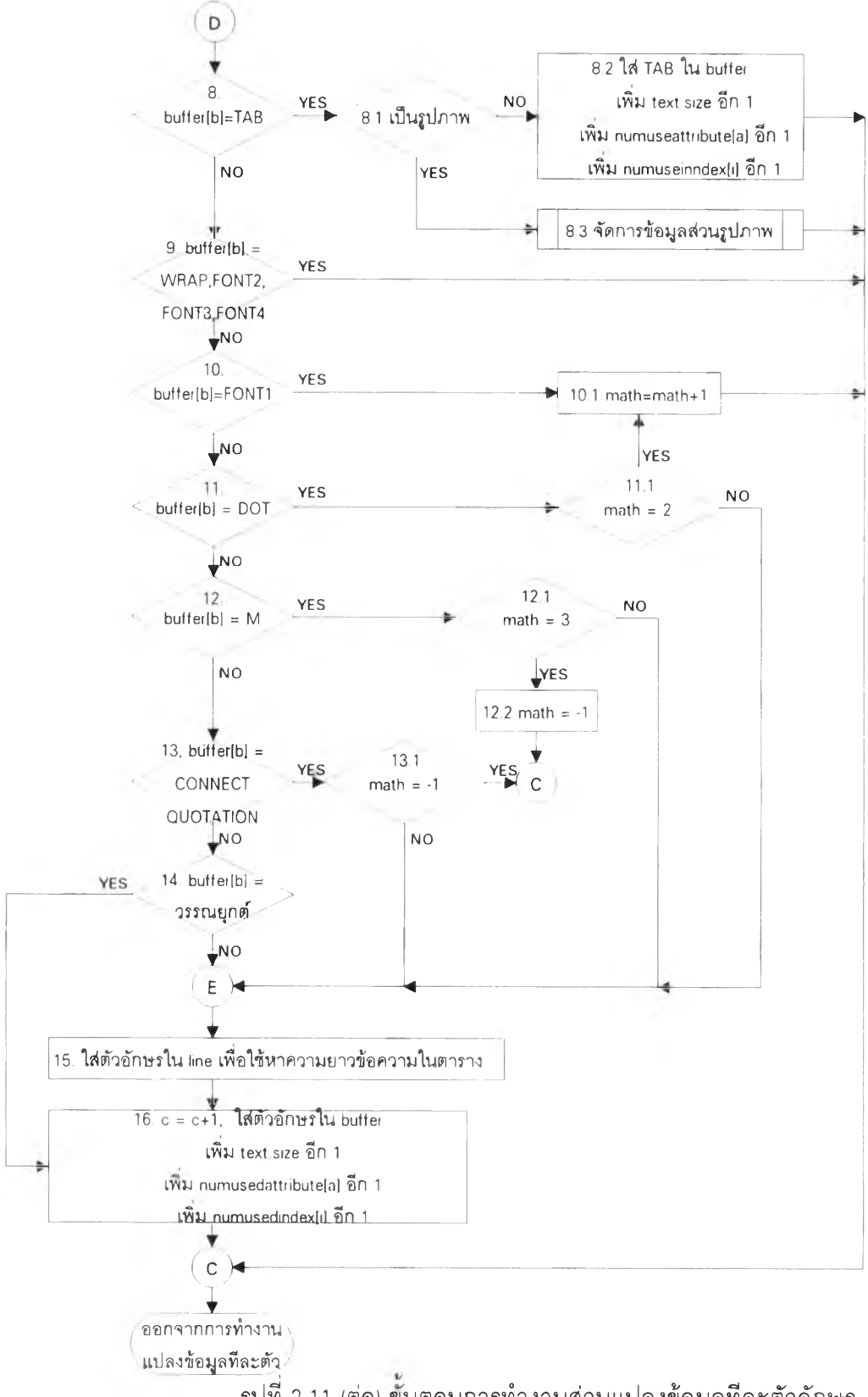
1. ถ้าตัวอักษรลำดับที่ b ใน buffer เป็น CR แสดงว่าเป็นรหัสขึ้นบรรทัดใหม่ จะทำงานดังนี้

1.1 ถ้า table เป็น 0 แสดงว่าไม่ได้สร้างตารางอยู่ จะทำข้อต่อไป ถ้า table ไม่เป็น 0 แสดงว่าสร้างตารางอยู่จะข้ามไปข้อ 1.4

1.2 ใส่รหัส CR ใน buffer พร้อมทั้งเพิ่มขนาดส่วน Text เพิ่มจำนวนตัวอักษรใน numusedindex และ numusedattribute



รูปที่ 2.11 ขั้นตอนการทำงานส่วนแปลงข้อมูลทีละตัวอักษร



รูปที่ 2.11 (ต่อ) ขั้นตอนการทำงานส่วนแปลงข้อมูลทีละตัวอักษร

1.3 กำหนดให้ c เป็น 0 ซึ่งเป็นคอลัมน์ของตาราง กำหนดให้ math เป็น 1 ใช้ตรวจสอบสมการคณิตศาสตร์ และลบข้อมูลใน line ที่ใช้หาความยาวข้อความ จากนั้นออกจากการทำงานส่วนแปลงข้อมูล

1.4 ถ้า table เป็น 2 แสดงว่าบรรทัดนั้นไม่มีรหัสตารางจึงข้ามไปข้อ 1.6 ถ้า table ไม่เป็น 2 แสดงว่าบรรทัดนั้นมีรหัสตารางจะทำข้อต่อไป

1.5 กำหนดให้ table เป็น 2 ใช้ตรวจสอบรหัสตาราง และหาความยาวของข้อความในบรรทัดนั้นเพื่อปรับความสูงบรรทัดของตาราง จากนั้นทำต่อข้อ 1.2, ข้อ 1.3

1.6 เมื่อบรรทัดนั้นไม่มีรหัสตารางแล้วจะสร้างรูปวาดเส้นตรงของตาราง (รายละเอียดดูเพิ่มเติมในผังงานย่อยแสดงดังรูปที่ 2.13) และกำหนดให้ table เป็น 0 เพื่อแสดงว่าไม่ได้สร้างตารางอยู่ จากนั้นทำต่อข้อ 1.2, ข้อ 1.3

2. ถ้าตัวอักษรลำดับที่ b ใน buffer เป็น WRAP แสดงว่าเป็นรหัสตัดคำซึ่งไม่มีผลต่อการแปลงข้อมูล จึงเพิ่ม b อีก 2 เพื่อข้ามรหัส 0x0D 0x0A จากนั้นออกจากการทำงานส่วนแปลงข้อมูล

3. ถ้าตัวอักษรลำดับที่ b ใน buffer เป็น BOLD, ITALIC, ENLARGE, SUPERSCRIPT, SUBSCRIPT เป็นรหัสรูปแบบพิเศษที่กำกับปิดหัวท้ายข้อความ ซึ่งมีผลต่อส่วน Attribute และส่วน Index จะทำงานดังนี้

3.1 ถ้า numusedattribute มากกว่า 0 แสดงว่ารหัสรูปแบบพิเศษนี้กำกับท้ายข้อความจะทำข้อต่อไป ถ้า numusedattribute เป็น 0 แสดงว่ารหัสรูปแบบพิเศษนี้กำกับหัวข้อความจะข้ามไปข้อ 3.3

3.2 จัดเก็บ attribute และ numusedattribute ลงใน attributeinfo พร้อมทั้งเพิ่มขนาดส่วน Attribute

3.3 นำรหัสรูปแบบพิเศษไปกำหนดบิตลักษณะประจำใน attribute โดยใช้ XOR จากนั้นข้ามไปข้อ 4.1 เพื่อทำงานต่อในส่วน Index

4. ถ้าตัวอักษรลำดับที่ b ใน buffer เป็น UNDERLINE, DOUBLELINE เป็นรหัสรูปแบบพิเศษที่กำกับปิดหัวท้ายข้อความ ซึ่งมีผลต่อส่วน Index จะทำงานดังนี้

4.1 ถ้า numusedindex มากกว่า 0 แสดงว่ารหัสรูปแบบพิเศษนี้กำกับท้ายข้อความจะทำข้อต่อไป ถ้า numusedindex เป็น 0 แสดงว่ารหัสรูปแบบพิเศษนี้กำกับหัวข้อความจะข้ามไปข้อ 4.3

4.2 จัดเก็บ index และ numusedindex ลงใน indexinfo พร้อมทั้งเพิ่ม

ขนาดส่วน Index

4.3 นำรหัสรูปแบบพิเศษไปกำหนดบิตดรรชนีใน index โดยใช้ XOR จากนั้นออกจากการทำงานส่วนแปลงข้อมูล

5. ถ้าตัวอักษรลำดับที่ b ใน buffer เป็น UL, UM, UR, ML, MM, BL, BM, H เป็นรหัสตาราง จะทำงานดังนี้

5.1 ถ้า table เท่ากับ 1 แสดงว่าพบรหัสตารางตัวแรกจึงต้องสร้างตารางในข้อต่อไป ถ้า table ไม่เท่ากับ 1 แสดงว่าไม่ใช่รหัสตารางตัวแรกจะข้ามไปข้อ 5.3

5.2 เมื่อพบรหัสตารางตัวแรกจึงต้องหาคอลัมน์และแถวของรหัสตาราง เพื่อสร้างเส้นตรงก่อน นอกจากนี้จุฬาจารึกใช้ตัวอักษรมีความกว้างไม่คงที่ จึงต้องหาความกว้างของแต่ละคอลัมน์ด้วย และจุฬาจารึกสามารถใช้ตัวอักษรหลายขนาดในเอกสารเดียวกันจึงต้องหาความสูงของแต่ละบรรทัดในตารางด้วย (รายละเอียดดูเพิ่มเติมในผังงานย่อยแสดงดังรูปที่ 2.12) จากนั้นกำหนดให้ table เท่ากับ 2 แสดงว่ากำลังสร้างตารางอยู่

5.3 พบรหัสตารางจึงเพิ่ม table อีก 1 และเพิ่มคอลัมน์ c อีก 1 จากนั้นออกจากการทำงานส่วนแปลงข้อมูล

6. ถ้าตัวอักษรลำดับที่ b ใน buffer เป็น MR, V เป็นรหัสตาราง ทำงานดังนี้

6.1 ถ้า table เท่ากับ 1 แสดงว่าพบรหัสตารางตัวแรกจึงต้องสร้างตารางในข้อต่อไป ถ้า table ไม่เท่ากับ 1 แสดงว่าไม่ใช่รหัสตารางตัวแรกจะข้ามไปข้อ 6.3

6.2 เมื่อพบรหัสตารางตัวแรกจึงต้องหาคอลัมน์และแถวของรหัสตาราง เพื่อสร้างเส้นตรงก่อน นอกจากนี้จุฬาจารึกใช้ตัวอักษรมีความกว้างไม่คงที่ จึงต้องหาความกว้างของแต่ละคอลัมน์ด้วย และจุฬาจารึกสามารถใช้ตัวอักษรหลายขนาดในเอกสารเดียวกันจึงต้องหาความสูงของแต่ละบรรทัดในตารางด้วย (รายละเอียดดูเพิ่มเติมในผังงานย่อยแสดงดังรูปที่ 2.12) จากนั้นกำหนดให้ table เท่ากับ 2 แสดงว่ากำลังสร้างตารางอยู่

6.3 เนื่องจากตัวอักษรมีความกว้างไม่คงที่จึงต้องหาความยาวข้อความแล้วเติมช่องว่างให้ข้อความในตารางจนมีความยาวใกล้เคียงกัน จากนั้นทำต่อข้อ 5.3

7. ถ้าตัวอักษรลำดับที่ b ใน buffer เป็น BR ซึ่งรหัสตารางตัวสุดท้ายจะสร้างรูปวาดเส้นตรงของตาราง (รายละเอียดดูเพิ่มเติมในผังงานย่อยแสดงดังรูปที่ 2.13) และกำหนดให้ table เป็น 0 เพื่อแสดงว่าไม่ได้สร้างตารางอยู่ จากนั้นออกจากการทำงานส่วนนี้

8. ถ้าตัวอักษรลำดับที่ b ใน buffer เป็น TAB อาจเป็นรหัสตั้งระยะหรือตัวแรกของรหัสรูปภาพ ทำงานดังนี้

8.1 ตรวจว่าข้อมูลต่อไปเป็นรหัส 0x0D, 0x0A และ picnode หรือไม่ ถ้าคำตอบคือ ใช่ แสดงว่าข้อมูลต่อไปเป็นรูปภาพ ข้ามไปทำข้อ 8.3 ถ้าคำตอบคือ ไม่ใช่ แสดงว่ารหัสนี้เป็นรหัสตั้งระยะ ทำข้อต่อไป

8.2 ใส่รหัส TAB ใน buffer พร้อมทั้งเพิ่มขนาดส่วน Text เพิ่มจำนวนตัวอักษรใน numusedindex และ numusedattribute จากนั้นออกจากการทำงานส่วนแปลงข้อมูล

8.3 ข้อมูลต่อไปเป็นรูปภาพจึงต้องจัดการข้อมูลส่วนรูปภาพ (รายละเอียดดูเพิ่มเติมในผังงานย่อยแสดงดังรูปที่ 2.14) จากนั้นออกจากการทำงานส่วนแปลงข้อมูล

9. ถ้าตัวอักษรลำดับที่ b ใน buffer เป็น WRAP77, FONT2, FONT3, FONT4 ซึ่งไม่มีผลต่อการแปลงข้อมูล จึงออกจากการทำงานส่วนแปลงข้อมูล

10. ถ้าตัวอักษรลำดับที่ b ใน buffer เป็น FONT1 แสดงว่าเป็นแฟ้มข้อมูลเวิร์ดจุฬารุ่นที่ 1.6 อาจมีสมการคณิตศาสตร์จึงเพิ่ม math อีก 1 จากนั้นออกจากการทำงานส่วนนี้

11. ถ้าตัวอักษรลำดับที่ b ใน buffer เป็น DOT อาจเป็นคำสั่งจุดของสมการคณิตศาสตร์หรือตัวอักษรจุด โดยตรวจสอบ math เป็น 2 หรือไม่ ถ้าคำตอบคือ ใช่ แสดงว่าเป็นคำสั่งจุดของสมการคณิตศาสตร์ จึงเพิ่ม math อีก 1 จากนั้นออกจากการทำงานส่วนแปลงข้อมูล ถ้าคำตอบคือ ไม่ใช่ แสดงว่าเป็นตัวอักษรจุดจึงข้ามไปทำข้อ 15

12. ถ้าตัวอักษรลำดับที่ b ใน buffer เป็น M อาจเป็นคำสั่งจุดของสมการคณิตศาสตร์หรือตัวอักษร M โดยตรวจสอบ math เป็น 3 หรือไม่ ถ้าคำตอบคือ ใช่ แสดงว่าเป็นคำสั่งจุดของสมการคณิตศาสตร์ จึงกำหนดให้ math เป็น -1 จากนั้นออกจากการทำงานส่วนแปลงข้อมูล ถ้าคำตอบคือ ไม่ใช่ แสดงว่าเป็นตัวอักษร M จึงข้ามไปทำข้อ 15

13. ถ้าตัวอักษรลำดับที่ b ใน buffer เป็น CONNECT, QUOTATION ตรวจสอบ math เป็น -1 หรือไม่ ถ้าคำตอบคือ ใช่ แสดงว่าบรรทัดนี้เป็นสมการคณิตศาสตร์ จะข้ามตัวอักษรนี้ไป จากนั้นออกจากการทำงานส่วนแปลงข้อมูล ถ้าคำตอบคือ ไม่ใช่ แสดงว่าเป็นตัวอักษรจึงข้ามไปทำข้อ 15

14 ถ้าตัวอักษรลำดับที่ b ใน buffer เป็นวรรณยุกต์จะไม่จัดเก็บตัวอักษรเหล่านี้ใน line ซึ่งใช้หาความยาวข้อความ จะข้ามไปข้อ 16

15. จัดเก็บตัวอักษรใน line เพื่อใช้หาความยาวข้อความ

16 เพิ่มคอลัมน์ c อีก 1 และใส่ตัวอักษรใน buffer พร้อมทั้งเพิ่มขนาดส่วน

Text เพิ่มจำนวนตัวอักษรใน numusedindex และ numusedattribute จากนั้นออกจากการทำงานส่วนแปลงข้อมูล

2.3 ส่วนหาคอลัมน์และแถวสำหรับสร้างเส้นตรงของตาราง เมื่อพบรหัสตารางตัวแรกในการแปลงข้อมูลทีละตัวนั้นจำเป็นต้องหาคอลัมน์และแถวของรหัสตารางเพื่อสร้างเส้นตรงก่อน และหาความกว้างของแต่ละคอลัมน์และความสูงของแต่ละบรรทัดในตารางด้วย ซึ่งลำดับที่ b ใน buffer จะยังไม่เปลี่ยนแปลงจนกว่าออกจากการทำงานส่วนนี้ จึงใช้ตัวชี้ ptr ค้นหารหัสตารางแทน และใช้ตัวแปร j ควบคุมการค้นหารหัสตาราง โดยกำหนดให้ j เป็น 0 เมื่อไม่มีรหัสตารางต่อไป และยังมีแถวลำดับของ col จัดเก็บความกว้างของคอลัมน์ แถวลำดับของ row จัดเก็บความสูงของบรรทัดในตาราง แถวลำดับของ ver จัดเก็บคอลัมน์ c และแถว r ของเส้นตรงแนวตั้ง และแถวลำดับของ hor จัดเก็บคอลัมน์ c และแถว r ของเส้นตรงแนวนอน การทำงานส่วนนี้จะตรวจสอบตัวอักษรที่ตัวชี้ ptr ว่าเป็นรหัสใดแล้วแยกไปทำงานแต่ละขั้นตอนต่อไป ผังงานย่อยแสดงดังรูปที่ 2.12 มีขั้นตอนการทำงานดังนี้

1. กำหนดให้ตัวควบคุมการค้นหารหัสตาราง j เป็น 2 กำหนดให้แถว r เป็น 0 กำหนดให้ความกว้างคอลัมน์ใน col เป็น 0 และความสูงของบรรทัดในตารางเป็น NmLineHigh ซึ่งเป็นความสูงของตัวอักษรปกติ

2. ถ้า j เท่ากับ 0 แสดงว่าไม่พบรหัสตารางต่อไปจึงออกจากการทำงานส่วนหาคอลัมน์และแถวของตาราง ถ้า j ไม่เท่ากับ 0 แสดงว่ายังมีรหัสตารางต่อไปจะทำงานข้อต่อไป

3. ถ้าตัวอักษรในตัวชี้ ptr เป็น CR ซึ่งเป็นรหัสขึ้นบรรทัดใหม่จะตรวจสอบตัวควบคุมการค้นหารหัสตาราง j

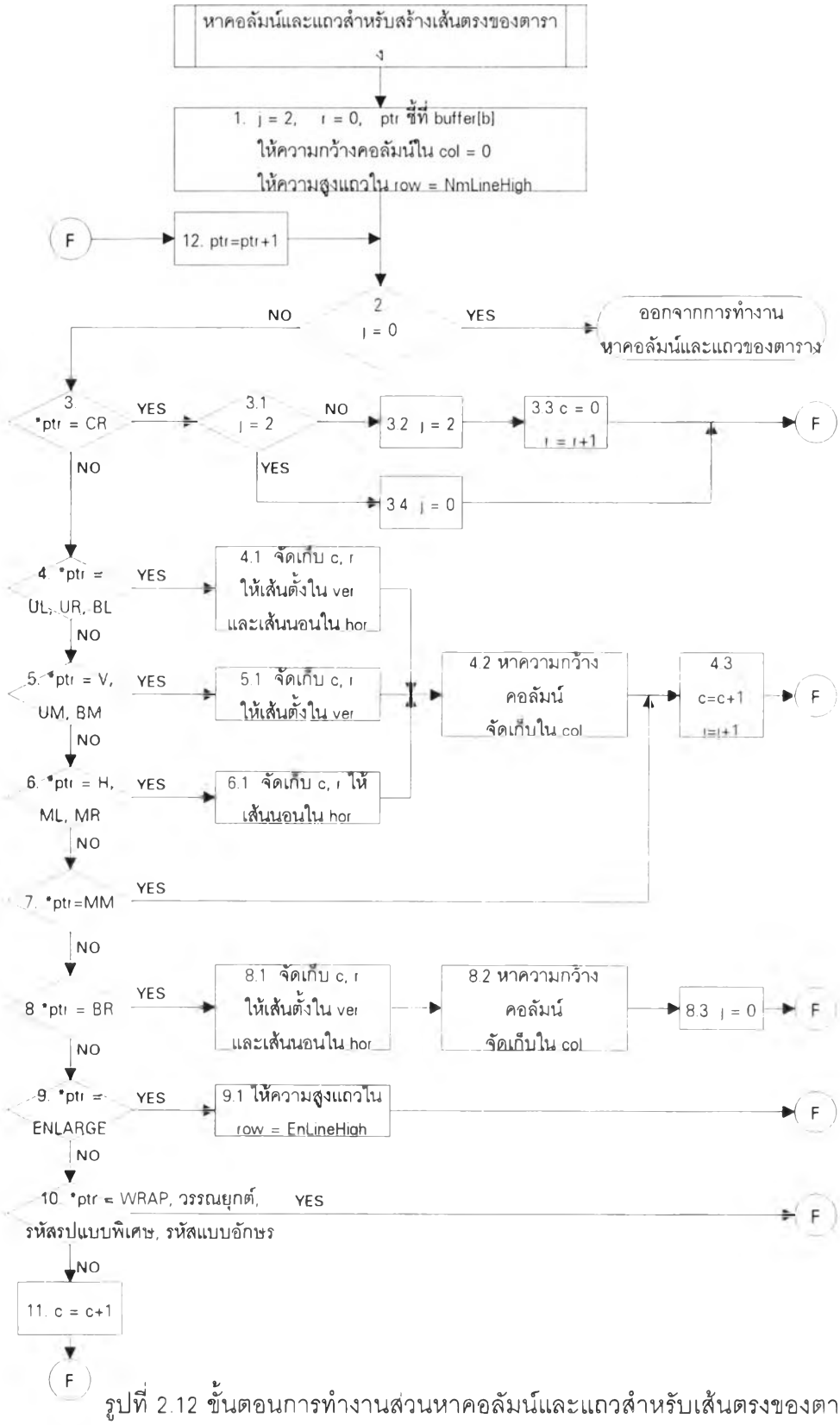
ถ้า j เป็น 2 แสดงว่าบรรทัดนั้นไม่มีรหัสตารางจะกำหนดให้ j เป็น 0 เมื่อย้อนกลับไปทำงานข้อ 2 จะออกจากการทำงานส่วนหาคอลัมน์และแถว

ถ้า j ไม่เป็น 2 แสดงว่าบรรทัดนั้นยังมีรหัสตารางจะกำหนดให้ j เป็น 2 ให้คอลัมน์ c เป็น 0 และเพิ่มแถว r อีก 1 จากนั้นข้ามไปทำต่อข้อ 12

4 ถ้าตัวอักษรในตัวชี้ ptr เป็น UL, UR, BL ซึ่งเป็นรหัสตารางที่สร้างเส้นตรงแนวตั้งและแนวนอน จะทำงานดังนี้

4.1 จัดเก็บคอลัมน์ c และแถว r เพื่อสร้างเส้นตรงแนวตั้งในแถวลำดับ ver และจัดเก็บคอลัมน์ c และแถว r เพื่อสร้างเส้นตรงแนวนอนในแถวลำดับ hor

4.2 หาความกว้างของคอลัมน์ col โดยเปรียบเทียบความยาวข้อความ



รูปที่ 2.12 ขั้นตอนการทำงานส่วนหาคอลัมน์และแถวสำหรับเส้นตรงของตาราง

กับค่าความกว้างคอลัมน์ที่จัดเก็บใน col ถ้าความยาวข้อความมากกว่าจะจัดเก็บความยาวข้อความนั้นเป็นความกว้างคอลัมน์แทน

4.3 เพิ่มคอลัมน์ c อีก 1 และเพิ่มตัวควบคุมการค้นหารหัสตาราง j อีก 1 จากนั้นข้ามไปทำต่อข้อ 12

5. ถ้าตัวอักษรในตัวชี้ ptr เป็น V, UM, BM ซึ่งเป็นรหัสตารางที่สร้างเส้นตรงแนวตั้ง จึงจัดเก็บคอลัมน์ c และแถว r ในแถวลำดับ ver จากนั้นทำต่อข้อ 4.2

6. ถ้าตัวอักษรในตัวชี้ ptr เป็น H, ML, MR ซึ่งเป็นรหัสตารางที่สร้างเส้นตรงแนวนอน จึงจัดเก็บคอลัมน์ c และแถว r ในแถวลำดับ hor จากนั้นทำต่อข้อ 4.2

7. ถ้าตัวอักษรในตัวชี้ ptr เป็น MM ซึ่งเป็นรหัสตารางที่เป็นจุดตัดของเส้นนอนและเส้นตั้ง จึงไม่มีผลต่อการจัดเก็บคอลัมน์ c, แถว r และความกว้างคอลัมน์ จากนั้นทำต่อข้อ 4.3

8. ถ้าตัวอักษรในตัวชี้ ptr เป็น BR ซึ่งเป็นรหัสตารางที่สร้างเส้นตรงแนวตั้งและแนวนอน จึงจัดเก็บคอลัมน์ c และแถว r ในแถวลำดับ ver และในแถวลำดับ hor จากนั้นหาความกว้างของคอลัมน์ col แล้วเปรียบเทียบกับความกว้างคอลัมน์และจัดเก็บใน col เนื่องจาก BR เป็นรหัสตารางตัวสุดท้ายจึงกำหนดให้ตัวควบคุมการค้นหารหัสตาราง j เป็น 0 เมื่อย้อนกลับไปทำข้อ 2 จะออกจากการทำงานส่วนหาคอลัมน์และแถว

9. ถ้าตัวอักษรในตัวชี้ ptr เป็น ENLARGE ซึ่งเป็นรหัสรูปแบบพิเศษตัวใหญ่ แสดงว่าบรรทัดนั้นในตารางต้องใช้ตัวอักษรตัวใหญ่ จึงให้ความสูงของบรรทัดนั้นเป็นความสูงของตัวอักษรใหญ่

10. ถ้าตัวอักษรในตัวชี้ ptr เป็น WRAP, วรรณยุกต์, รหัสรูปแบบพิเศษ, รหัสแบบอักษร ซึ่งไม่มีผลต่อการหาคอลัมน์และแถว จึงข้ามไปทำต่อข้อ 12

11. ตัวอักษรในตัวชี้ ptr เป็นตัวอักษรที่มีผลต่อคอลัมน์ของตาราง จึงเพิ่มคอลัมน์ c อีก 1 แล้วทำข้อต่อไป

12. เพิ่ม ptr อีก 1 เพื่อใช้ตัวอักษรตัวถัดไป แล้วย้อนกลับไปทำต่อข้อ 2

2.4 <u>ส่วนสร้างเส้นตรงของตาราง</u> เมื่อพบรหัสตารางในการทำงานส่วนแปลงข้อมูลทีละตัวนั้นจำเป็นต้องหาคอลัมน์และแถวของรหัสตารางเพื่อสร้างเส้นตรงก่อน โดยจัดเก็บคอลัมน์และแถวของเส้นตั้งในแถวลำดับ ver และคอลัมน์และแถวของเส้นนอนในแถวลำดับ hor พร้อมทั้งหาความกว้างของแต่ละคอลัมน์และความสูงของแต่ละบรรทัดในตารางด้วย หลังจากนั้นหาความยาวข้อความในตารางแล้วเติมช่องว่างจนข้อความเหล่านั้นมีความยาวใกล้เคียงกัน

จากนั้นนำคอลัมน์และแถวของรหัสตารางมาสร้างเส้นตรงของตาราง การสร้างเส้นตรงของตาราง มีผลต่อส่วน Overlay และส่วน Section การทำงานส่วนนี้มีตัวแปร verline เก็บจำนวนเส้นตรงแนวตั้ง และตัวแปร horline เก็บจำนวนเส้นตรงแนวนอน ผังงานย่อยของการทำงานส่วนนี้แสดงดังรูปที่ 2.13 ขั้นตอนการทำงานดังนี้

1. กำหนดให้จำนวนเส้นตรงแนวตั้ง verline เป็น 0 และเส้นตรงแนวนอน horline เป็น 0

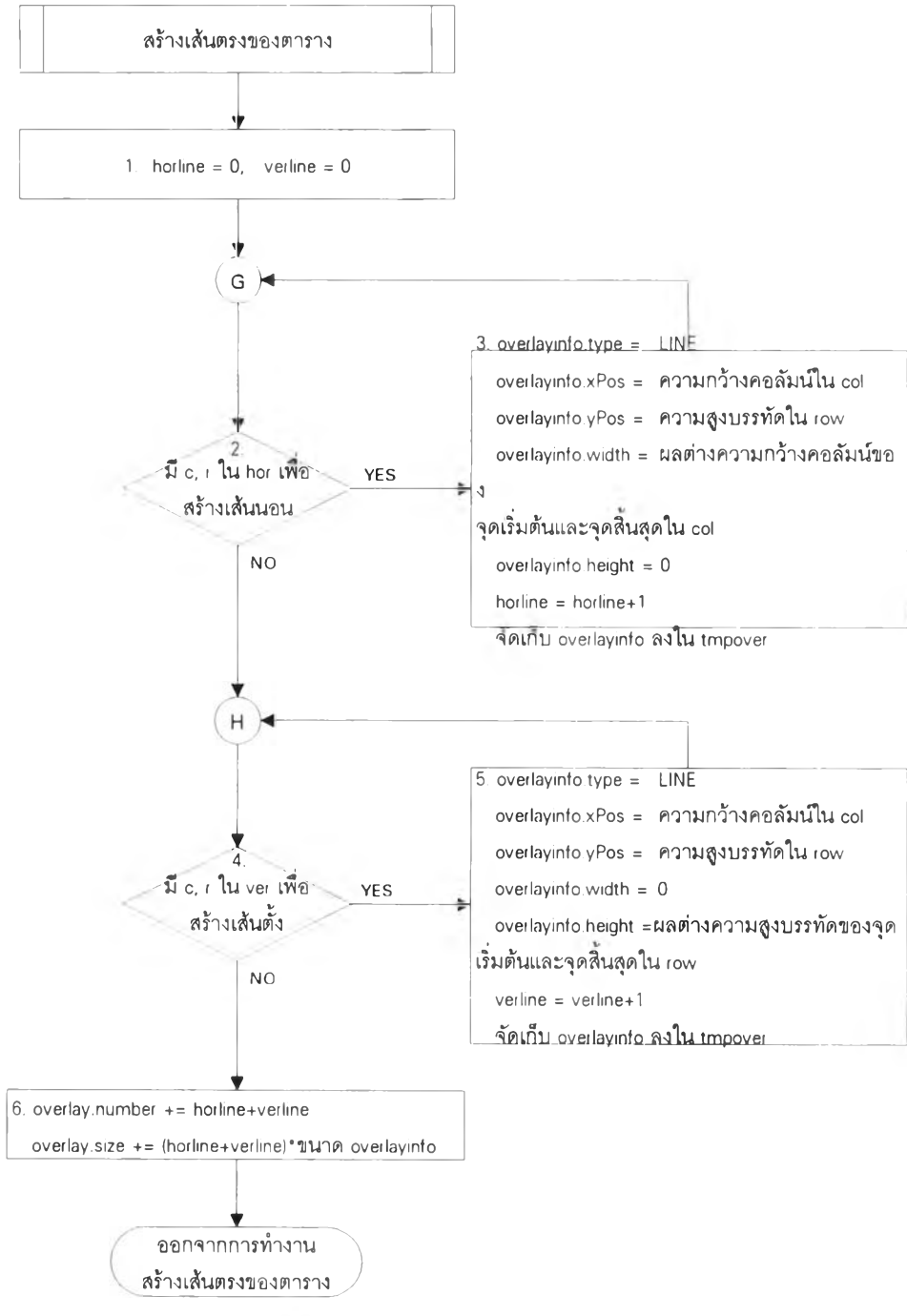
2. ตรวจสอบว่ามีคอลัมน์ c และแถว r ในแถวลำดับ hor เพื่อสร้างเส้นตรงแนวนอนหรือไม่ ถ้าคำตอบคือ ใช่ แสดงว่าต้องสร้างรูปวาดเส้นตรงแนวนอนทีละเส้น จะทำข้อต่อไป ถ้าคำตอบคือ ไม่ใช่ จะข้ามไปทำงานข้อ 4

3 สร้างเส้นตรงแนวนอน โดยให้ชนิดรูปวาด overlay.type เป็น LINE, ให้ตำแหน่งแนวนอน overlay.xPos เป็นความกว้างคอลัมน์ c ใน col, ให้ตำแหน่งแนวตั้ง overlay.yPos เป็นความสูงบรรทัดของแถว r ใน row, ให้ความกว้างของรูปวาด overlay.width เป็นผลต่างความกว้างคอลัมน์ของจุดเริ่มต้นและจุดสิ้นสุดใน col, ให้ความสูงของรูปวาด overlay.height เป็น 0,เพิ่มจำนวนเส้นตรงแนวนอน horline อีก 1 และจัดเก็บรูปวาด overlayinfo ลงแฟ้มข้อมูลชั่วคราว tmpover ย้อนกลับไปทำต่อข้อ 2

4 ตรวจสอบว่ามีคอลัมน์ c และแถว r ในแถวลำดับ ver เพื่อสร้างเส้นตรงแนวตั้งหรือไม่ ถ้าคำตอบคือ ใช่ ต้องสร้างรูปวาดเส้นตรงแนวตั้งทีละเส้น จะทำข้อต่อไป ถ้าคำตอบคือ ไม่ใช่ แสดงว่าจบขั้นตอนการสร้างเส้นตรงของตาราง จะข้ามไปทำต่อข้อ 6

5. สร้างเส้นตรงแนวตั้ง โดยให้ชนิดรูปวาด overlay.type เป็น LINE, ให้ตำแหน่งแนวนอน overlay.xPos เป็นความกว้างคอลัมน์ c ใน col, ให้ตำแหน่งแนวตั้ง overlay.yPos เป็นความสูงบรรทัดของแถว r ใน row, ให้ความกว้างของรูปวาด overlay.width เป็น 0, ให้ความสูงของรูปวาด overlay.height เป็นผลต่างความสูงของบรรทัดของจุดเริ่มต้นและจุดสิ้นสุดใน ver, เพิ่มจำนวนเส้นตรงแนวตั้ง verline อีก 1 และจัดเก็บรูปวาด overlayinfo ลงแฟ้มข้อมูลชั่วคราว tmpover ย้อนกลับไปทำต่อข้อ 4

6. เมื่อสร้างเส้นตรงแนวนอนและเส้นตรงแนวตั้งเสร็จแล้ว จัดเก็บ horline และ verline ใน overlay.number เพื่อเขียนจำนวนรูปวาดของย่อหน้านี้ใน numdrawinfo ของ bodyparagraphinfo และเพิ่มขนาดส่วน overlay จากนั้นออกจากการทำงานส่วนสร้างเส้นตรงของตาราง



รูปที่ 2.13 ขั้นตอนการทำงานส่วนสร้างเส้นตรงของตาราง

2.5 ส่วนจัดการข้อมูลส่วนรูปภาพ จุฬาจารึก 77 จัดเก็บรูปภาพในหน่วยคอลัมน์และแถว แต่จุฬาจารึก 78 จัดเก็บรูปภาพในหน่วย twip จำเป็นต้องเปลี่ยนหน่วยก่อน ซึ่งรายละเอียดได้กล่าวไว้ในบทที่ 2 การจัดการข้อมูลส่วนรูปภาพมีผลต่อส่วน Picture, ส่วน Text, ส่วน Index, และส่วน Attribute ผังงานย่อยของการทำงานส่วนนี้แสดงดังรูปที่ 2.14 ขั้นตอนการทำงานดังนี้

1. อ่านข้อมูล picnode และขนาดของรูปภาพ

2. ตรวจสอบว่าข้อมูล hdibmp ใน picnode ไม่เป็น null หรือไม่ ถ้าคำตอบคือ ใช่ แสดงว่าเป็นรูปภาพแบบ BMP จะทำข้อต่อไป ถ้าคำตอบคือ ไม่ใช่ แสดงว่าเป็นรูปภาพแบบ WMF จะข้ามไปทำต่อข้อ 4

3. อ่านข้อมูล bmiheader ซึ่งรูปภาพแบบ BMP ต้องใช้ข้อมูลใน bmiheader คำนวณเปลี่ยนหน่วยเป็น twip ในขณะที่รูปภาพแบบ WMF ใช้เพียงข้อมูลใน picnode เท่านั้น

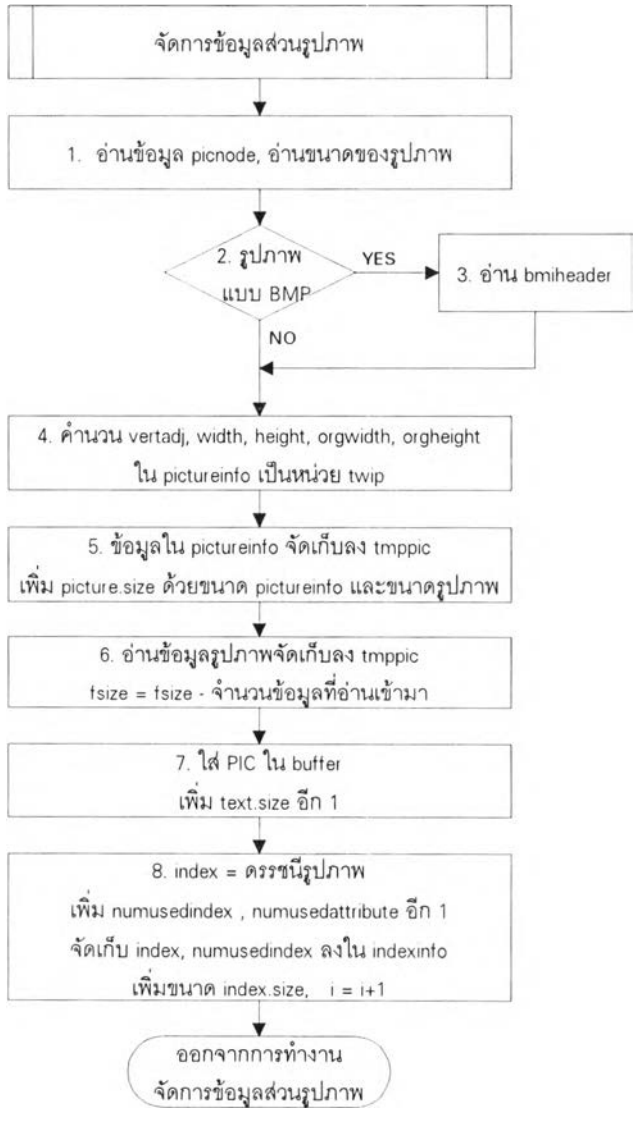
4. คำนวณรายละเอียดของรูปภาพใน pictureinfo เป็นหน่วย twip โดยรายละเอียดของการเปลี่ยนหน่วยกล่าวไว้ในบทที่ 2

5. จัดเก็บรายละเอียดของรูปภาพใน pictureinfo ลงในแฟ้มข้อมูลชั่วคราว tmppic จากนั้นเพิ่มขนาด pictureinfo และขนาดรูปภาพให้ส่วน Picture

6. อ่านข้อมูลรูปภาพจากแฟ้มข้อมูลเดิมแล้วจัดเก็บลงแฟ้มข้อมูลชั่วคราว tmppic จากนั้นต้องลดขนาดแฟ้มข้อมูลเดิม fsize ด้วยจำนวนข้อมูลที่อ่านเข้ามา

7. ใส่รหัส PIC ใน buffer ให้ส่วน Text และเพิ่มขนาดส่วน Text อีก 1

8. จัดเก็บดรรชนีรูปภาพใน index และเพิ่มจำนวนตัวอักษรใน numusedindex และ numusedattribute อีก 1 จากนั้นจัดเก็บ index และ numusedindex ลงใน indexinfo และเพิ่มขนาดส่วน index แล้วออกจากการจัดการข้อมูลส่วนรูปภาพ



รูปที่ 2.14 ขั้นตอนการทำงานส่วนจัดการข้อมูลรูปภาพ