บทที่ 2

แนวคิดและทฤษฎีที่ใช้

ในบทนี้จะกล่าวถึงแนวคิดและทฤษฎีที่เกี่ยวข้องที่จะนำมาใช้ในการแก้ไขปัญหา โดยจะกล่าวถึง องค์ประกอบต่างๆที่เกี่ยวข้องในการค้นข้อความในเอกสารพีดีเอฟ มาตรฐานการเข้ารหัสของแบบอักษรไทย และแนวคิดการค้นข้อความโดยวิธีการเปรียบเทียบสายอักขระ

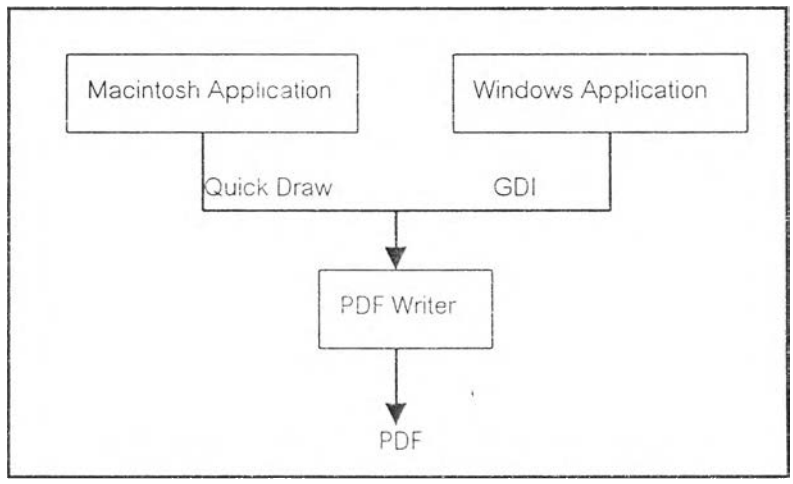
2.1 เอกสารพีดีเอฟ

ก่อนที่จะสร้างกระบวนการค้นข้อความไทยในเอกสารพีดีเอฟ ควรจะทราบถึงข้อกำหนดและแนวคิดของเอกสารพีดีเอฟ เช่น วิธีในการสร้างเอกสารพีดีเอฟ โครงสร้างแฟ้มเอกสารพีดีเอฟ ข้อกำหนดประเภทแบบอักษร ข้อกำหนดการเข้ารหัสของแบบอักษรที่ใช้ในเอกสารพีดีเอฟ

2.1.1 การสร้างเอกสารพีดีเอฟ

การสร้างเอกสารพีดีเอฟ[1] สามารถสร้างได้ 2 วิธี คือ

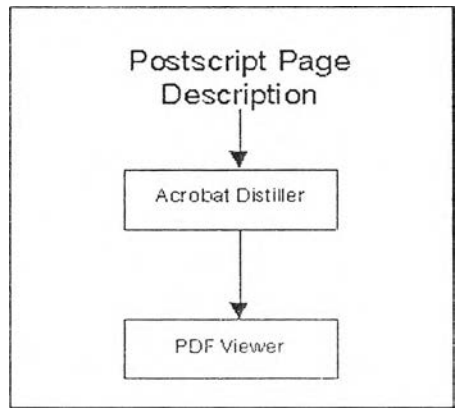
1. พีดีเอฟไรเตอร์ (PDF Writer®) สร้างเอกสารพีดีเอฟจากโปรแกรมสำเร็จประยุกต์ที่ผู้ใช้นิยมใช้อยู่โดยสั่งให้โปรแกรมสำเร็จประยุกต์นั้นพิมพ์เอกสารผ่านพีดีเอฟไรเตอร์ พีดีเอฟไรเตอร์จะทำการนำรายละเอียดการพิมพ์นั้นมาสร้างเป็นเอกสารพีดีเอฟ (ดูรูปที่ 3ประกอบ)



รูปที่ 3 กระบวนการสร้างเอกสารพีดีเอฟโดยวิธีพีดีเอฟไรเตอร์

[1] Adobe Systems Incorporated.. Portable Document Format Reference Manual. Adobe Systems, 1999, pp. 19-21

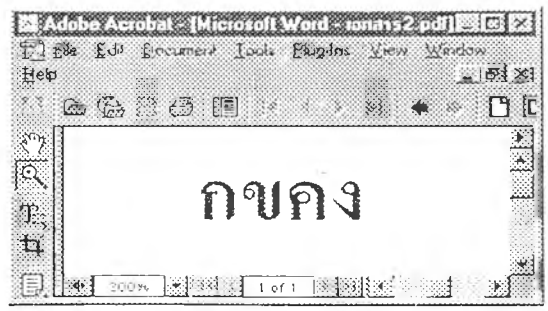
2. ดิสทิลเลอร์ (Distiller®) เป็นวิธีการสร้างเอกสารพีดีเอฟโดยจะทำการแปลงจากเอกสารโพสต์สคริปต์ (PostScript) มาเป็นเอกสารพีดีเอฟ เอกสารที่ประกอบด้วยข้อความทั่วๆไป ในการสร้างเป็นเอกสารพีดีเอฟสามารถทำได้ทั้ง 2 วิธี แต่ในเอกสารที่มีรูปภาพต้องการคุณภาพของรูปรูปที่จะแสดงได้สวยงาม ควรเลือกใช้วิธีดิสทิลเลอร์ การสร้างเอกสารพีดีเอฟโดยวิธีนี้จะได้เอกสารที่มีคุณภาพดีกว่าโดยวิธีพีดีเอฟไรเตอร์ เนื่องจากผู้ใช้จะสร้างเอกสารให้อยู่ในรูปแบบเอกสารโพสต์สคริปต์ก่อน แล้วจึงใช้ดิสทิลเลอร์ทำการแปลงเอกสารนั้นเป็นเอกสารพีดีเอฟ ซึ่งในการสร้างเอกสารโพสต์สคริปต์นั้น ผู้ใช้สามารถเขียนเอกสารขึ้นเองโดยใช้คำสั่งภาษาโพสต์สคริปต์ หรือจะสร้างจากโปรแกรมสำเร็จประยุกต์ที่ผู้ใช้ใช้งานอยู่ โดยการพิมพ์ผ่านตัวขับเครื่องพิมพ์โพสต์สคริปต์ (Postscript Printer Driver) เพื่อให้ได้แฟ้มเอกสารโพสต์สคริปต์ก่อนแล้วจึงทำการสร้างเอกสารพีดีเอฟโดยดิสทิลเลอร์ (ดูรูปที่ 4 ประกอบ)



รูปที่ 4 กระบวนการสร้างเอกสารพีดีเอฟโดยวิธีดิสทิลเลอร์

2.1.2 โครงสร้างแฟ้มเอกสารพีดีเอฟ

หัวข้อนี้จะแสดงให้ทราบถึงโครงสร้างแฟ้มเอกสารพีดีเอฟมีส่วนประกอบต่างๆอย่างไรบ้าง



รูปที่ 5 ตัวอย่างเอกสารพีดีเอฟแสดงข้อความ "กขคง"

```
%PDF-1 2
2 0 obj
<< /Filter /FlateDecode /Length 122 >>
stream
H00T{T0W0000d"ข05000จUD<ศk#0ะhKญ"0,0H0'X0|ต 0-Yt:!ขง
VD"# 0S000H)ส"000=(0ฒิ อภๆaj,{0ลa!0000}0ภษ`c~ๆ0ฟ0`฿`0ฐ+X
0v0c~0{0ฝยาเ0๑)๔,\0kq0-0ฎถ่ๆoL|+00
endstream
endobj
3 0 obj
<<
/ProcSet [/PDF /Text]
/Font <</F2 4 0 R>>
/ExtGState <</GS1 5 0 R>>
>>
endobj
7 0 obj
<<
/Type /Halftone
/Halftone Type1
/HalftoneName (Default)
/Frequency 60
/Angle 45
>>
endobj
5 0 obj
<<
/Type /ExtGState
/SA false
/SM 0.02
/TR /Identity
>>
endobj
4 0 obj
<<
/Type /Font
/Subtype /Type1
/FirstChar 1
/LastChar 4
/Widths [ 400 373 400 420  ]
/Encoding 10 0 R
/BaseFont /CLADOF+TTEBo00
/FontDescriptor 8 0 R
>>
endobj
8 0 obj
<<
/Type /FontDescriptor
/Ascent 0
/CapHeight 0
/Descent 0
/Flags 4
/FontBBox [ -380 -253 780 693 ]
/FontName /CLADOF+TTEBo00
/ItalicAngle 0
/StemV 0
/CharSet (/afii59681;/afii59682;/afii59684;/afii59687;)
/FontFile3 9 0 R
>>
endobj
10 0 obj
<<
/Type /Encoding
/Differences[
1/afii59681/afii59682;/afii59684;/afii59687;]
>>
endobj
9 0 obj
<< /Filter /FlateDecode /Length 15652 /Subtype
/Type1C >>
stream
H%oŒT{T0W0000d"ข05000จUD<ศk#0ะhKญ"0,0H0'X0|ต 0-Yt:!
ขงVD"# 0S000H)ส"000=(0ฒิGjฎ[๔0q0ส]฿=3gฮ7wพ฿๑0พ฿|๗ว
100รq7zขwc0ๆ00Em"H0-$/6{๙S0ฟ0ขผดQ0"๔t06čขW00(๐0
๔๙๑3๑;๑'G๐ผX0ผฉผ00|000๑015ฆง+4*0Rc~6RIF"X00ฆๆB๗000
ใ0ลฉ00"จฆ0ฦ7 hฆ0[EPy0ฮบภ0V8ฝ0ฮ-06ว1;nGฎฅน0ปฏ
o5vwu>ny๐ใ`,๗"ฝ0—ช&000ฆ
endstream
endobj
1 0 obj
<<
/Type /Page
/Parent 6 0 R
/Resources 3 0 R
/Contents 2 0 R
>>
endobj
6 0 obj
<<
/Type /Pages
/Kids [ 1 0 R ]
/Count 1
>>
endobj
11 0 obj
<<
/Type /Catalog
/Pages 6 0 R
>>
endobj
12 0 obj
<<
/CreationDate (D:19991019211628)
/Producer (Acrobat Distiller 4.0 for Windows)
/ModDate (D:19991019211643+07'00')
>>
endobj
xref
0 13
0000000000 65535 f
0000000604 00000 n
0000000816 00000 n
0000000966 00000 n
0000001098 00000 n
0000001277 00000 n
0000001720 00000 n
0000017464 00000 n
0000018545 00000 n
0000019699 00000 n
0000020179 00000 n
0000000657 00000 n
0000000796 00000 n
trailer
<<
/Size 13
/Info 12 0 R
/Root 11 0 R
/Prev 20458
/ID[<d53042f81477fcc8b4158fe1f5282832>]
>>
startxref
2223
%%EOF
```

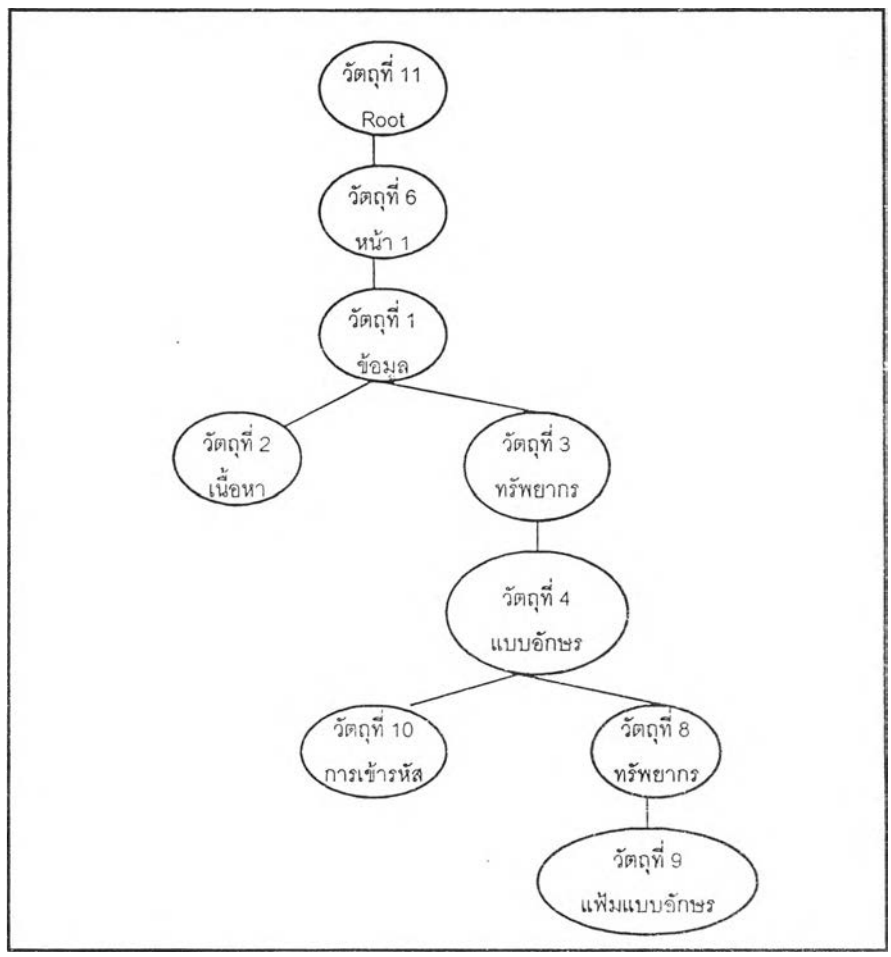
รูปที่ 6 ตัวอย่างโครงสร้างแฟ้มเอกสารพีดีเอฟแสดงข้อความ "กขคง"

จากรูปที่ 5 และรูปที่ 6 จะแสดงตัวอย่างเอกสารพีดีเอฟและโครงสร้างภายในของแฟ้มเอกสาร ว่าแฟ้มเอกสารพีดีเอฟนี้มีโครงสร้างแฟ้มอย่างไร เอกสารจะถูกนำมาแสดงได้อย่างไร

โครงสร้างแฟ้มเอกสารพีดีเอฟ[1] ประกอบด้วย 4 ส่วนด้วยกัน ดังนี้

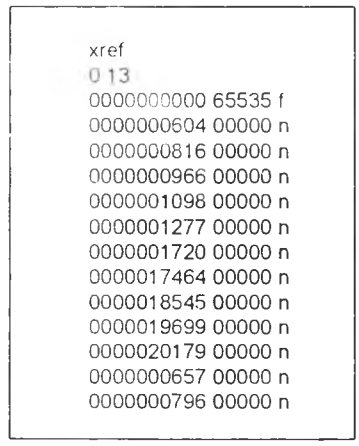
ส่วนที่หนึ่ง คือ ส่วนหัว(Header) ใช้ในการระบุว่า เอกสารพีดีเอฟนี้เป็นไปตามข้อกำหนดรุ่นที่เท่าไหร่ ดังจะเห็นได้จากตัวอย่างในรูปที่ 6 ในบรรทัดแรกระบุ %PDF-1.2 ซึ่งหมายถึง โครงสร้างแฟ้มเอกสารพีดีเอฟนี้เป็นไปตามข้อกำหนดรุ่นที่ 1.2

ส่วนที่สอง คือ ส่วนที่ใช้ในการแสดงเอกสาร(Body) จะประกอบด้วยวัตถุต่างๆที่ใช้ในการแสดงข้อความ รูปภาพ เสียง ภาพเคลื่อนไหว และข้อมูลอื่นๆ ในส่วนนี้จะถูกนำมาใช้ในการแสดงเอกสารพีดีเอฟ วัตถุต่างๆในส่วนที่สองนี้สัมพันธ์กันในลักษณะของโครงสร้างข้อมูลแบบต้นไม้ ข้อมูลบางส่วน เช่นเนื้อหาในเอกสารหรือแฟ้มแบบอักษรที่ผนวกมากับเอกสาร โดยปกติจะถูกบีบอัดข้อมูลไว้เพื่อลดขนาดของข้อมูล ในการใช้วัตถุในเอกสารต้องทำการขยายข้อมูลที่ถูกบีบอัดไว้เสียก่อนจึงเข้าใช้วัตถุต่างๆในส่วนนี้ได้



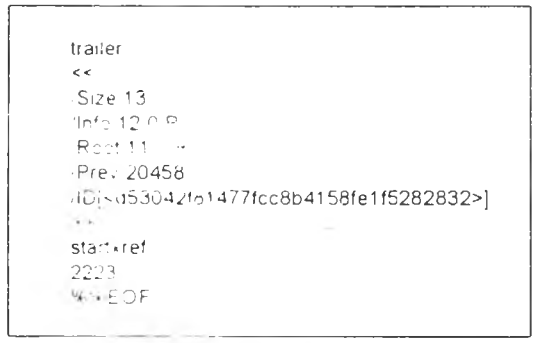
รูปที่ 7 โครงสร้างต้นไม้ของวัตถุที่ใช้ในการแสดงเอกสารพีดีเอฟ

ส่วนที่สาม คือ ตารางอ้างอิง(Cross Reference Table) เป็นข้อมูลตารางที่ช่วยให้เข้าใช้วัตถุในเอกสารพีดีเอฟได้โดยตรง เป็นวิธีการเข้าใช้แบบ Random Access ทำให้การอ่านข้อมูลในเอกสารพีดีเอฟทำได้รวดเร็ว ข้อมูลในส่วนนี้จะต้องเริ่มต้นด้วย xref บรรทัดถัดไปจะแสดงถึงจำนวนวัตถุที่มีอยู่ในเอกสารพีดีเอฟ ตัวอักษร f จะแสดงว่าวัตถุนั้นเป็นวัตถุอิสระหรือเป็นวัตถุเริ่มต้นของ link list ตัวอักษร n จะแสดงว่าวัตถุนั้นถูกใช้ในเอกสาร ตัวเลขในแต่ละแถวจะแสดงถึงตำแหน่งเริ่มต้นของวัตถุนั้นในแฟ้มเอกสารพีดีเอฟ (ดูรูปที่ 8 ประกอบ)



รูปที่ 8 ตัวอย่างตารางอ้างอิงในแฟ้มเอกสารพีดีเอฟ

ส่วนที่สี่ คือ ส่วนท้าย(Trailer) ในการอ่านแฟ้มเอกสารพีดีเอฟจะเริ่มอ่านที่ตอนท้ายของแฟ้มเอกสาร เพื่อให้ทราบตำแหน่งเริ่มต้นของตารางอ้างอิงเพื่อที่จะได้นำไปใช้ในการเข้าใช้วัตถุ ข้อมูลในส่วนท้ายนี้จะต้องเริ่มต้นด้วย trailer และปิดท้ายด้วย %%EOF บรรทัดก่อนที่จะระบุ %%EOF จะบอกถึงตำแหน่งเริ่มต้นของตารางอ้างอิงในแฟ้มเอกสารพีดีเอฟ นอกจากนี้ในส่วนนี้จะมีข้อมูลอื่นๆที่จะบอกให้ทราบถึง จำนวนวัตถุในแฟ้มเอกสารพีดีเอฟ วัตถุเริ่มต้น รายละเอียดในการสร้างเอกสารพีดีเอฟ เช่น ชื่อผู้สร้าง คำสำคัญ วันที่ และอื่นๆ (ดูรูปที่ 9 ประกอบ)



รูปที่ 9 ตัวอย่างข้อมูลส่วนท้ายของแฟ้มเอกสารพีดีเอฟ

2.1.3 แบบอักษรในเอกสารพีดีเอฟ[1]

โดยปกติเมื่อมีการแลกเปลี่ยนเอกสาร ผู้รับต้องมีแบบอักษรชุดเดียวกับที่ผู้ส่งใช้ในเอกสาร เพื่อที่ผู้รับจะได้เอกสารที่แสดงได้ตรงกับที่ผู้ใช้ส่งมา สำหรับในเอกสารพีดีเอฟแล้วจะใช้แบบอักษรเดียวกับที่ใช้ในเอกสารเมื่อมีแบบอักษรนั้นๆในเครื่องของผู้รับ แต่ถ้าไม่มีแบบอักษรนั้นจะนำแบบอักษรอื่นที่มีข้อมูลของแบบอักษรในลักษณะเดียวกันมาแสดงแทน แต่เพื่อความมั่นใจว่าเอกสารจะแสดงได้อย่างถูกต้อง ผู้สร้างเอกสารสามารถที่จะผนวกแบบอักษรที่ใช้ไปไว้ในเอกสาร วิธีนี้จะทำให้มั่นใจได้ว่าผู้รับจะได้เอกสารที่แสดงได้เหมือนกับที่ผู้สร้างสร้างเอกสาร แต่จะมีข้อเสียที่ทำให้ขนาดของเอกสารใหญ่ขึ้น

วัตถุที่ใช้ในการแสดงข้อความในเอกสารพีดีเอฟจะประกอบด้วย เนื้อหาที่ใช้ในการแสดงข้อความในเอกสารพีดีเอฟทรัพยากรที่ใช้ในการแสดงข้อความและแบบอักษร ข้อมูลแบบอักษรในเอกสารพีดีเอฟอาจมีเพียงชื่อแบบอักษรหรือมีแฟ้มชุดอักษรทั้งแฟ้มก็ได้ ขึ้นอยู่กับข้อกำหนดของผู้สร้างเอกสาร การจัดการแบบอักษรเป็นงานพื้นฐานที่สำคัญและถูกกล่าวถึงเสมอในข้อกำหนดเอกสารพีดีเอฟ

แบบอักษรเป็นข้อมูลที่สำคัญในการค้นข้อความในเอกสารพีดีเอฟ ข้อมูลแบบอักษรในเอกสารพีดีเอฟ เป็นข้อมูลชนิดพจนานุกรมที่เก็บข้อมูลหรือข้อกำหนดของแบบอักษรนั้นๆ เช่น ชนิดของแบบอักษร ชื่อของแบบอักษร การเข้ารหัสอักษร ข้อมูลที่ใช้ในการแสดงแบบอักษร หรือข้อมูลในการแสดงแทนเมื่อไม่มีแบบอักษรนั้นๆ

แบบอักษรที่ใช้ในเอกสารพีดีเอฟมีอยู่ด้วยกัน 4 ชนิด คือ

1. ประเภทที่ 0 (Type0)
2. ประเภทที่ 1 (Type1)
3. ประเภทที่ 3 (Type3)
4. ประเภททรูไทป์ (TrueType)

ประเภทที่ 0 (ในที่นี้จะขอกล่าวถึงเพียงสังเขป ทั้งนี้เนื่องจากไม่ได้ถูกนำมาใช้กับแบบอักษรไทยที่ใช้ในเอกสารพีดีเอฟ) เป็นแบบอักษรที่ถูกออกแบบมาเพื่อสนับสนุนสำหรับอักขระที่มีตัวอักษรเป็นจำนวนมาก เช่น ภาษาจีน ภาษาญี่ปุ่น ภาษาเกาหลี ซึ่งเป็นภาษาที่มีอักขระมากกว่า 256 อักขระหรือภาษาที่เป็นอักขระภาพ การเข้ารหัสตัวอักษรของแบบอักษรชนิดนี้จะใช้วิธีการพิเศษเพื่อที่จะสามารถเข้ารหัสอักขระจำนวนมากได้ โดยจะเก็บไว้ในข้อมูลชนิดหนึ่งที่เรียกว่าแผนที่อักษร

แบบอักษรประเภทที่ 1 มีขนาดเล็ก ให้คุณภาพตัวอักษรดี แม้ว่าจะต้องแสดงตัวอักษรขนาดเล็กบนอุปกรณ์แสดงผลที่มีรายละเอียดในการแสดงผลต่ำ เป็นแบบอักษรที่เหมาะสมที่สุดสำหรับภาษาโพสต์สคริปต์หรือเอกสารพีดีเอฟ ข้อกำหนดและรายละเอียดดังนี้

ตารางที่ 1 ข้อมูลส่วนต่างๆของแบบอักษรประเภทที่ 1

ข้อมูล	ชนิดของข้อมูล	ระบุถึง
Subtype	ชื่อ	ชนิดของแบบอักษร ต้องระบุเป็น Type1
BaseFont	ชื่อ	ชื่อของแบบอักษรโพสต์สคริปต์ที่ถูกกำหนดในแพ้มโพสต์สคริปต์สำหรับใช้ในการแสดงข้อความ เป็นชื่อที่จะถูกใช้ในการโปรแกรมภาษาโพสต์สคริปต์
FirstChar	จำนวนเต็ม	รหัสตัวอักษรตัวแรกที่ใช้ในแถวลำดับที่กำหนดความกว้างของอักขระที่ใช้ในเอกสาร
LastChar	จำนวนเต็ม	รหัสตัวอักษรตัวสุดท้ายที่ใช้ในแถวลำดับที่กำหนดความกว้างของอักขระที่ใช้ในเอกสาร
ความกว้าง	แถวลำดับ	ขนาดของแต่ละตัวอักษรที่ใช้ในเอกสารและจะระบุลำดับตัวอักษรตัวแรกและตัวอักษรตัวสุดท้ายที่ใช้ในแบบอักษร
Encoding	พจนานุกรม	การเข้ารหัสตัวอักษรของแบบอักษรที่ใช้ในเอกสาร
FontDescriptor	พจนานุกรม	ข้อมูลที่เป็นรายละเอียดต่างๆที่จะใช้ในการแสดงตัวอักษร เช่น ชื่อแบบอักษร อักขระในแบบอักษรที่ถูกใช้ในการแสดง กรอบขนาดในการแสดงแบบอักษร

```
14 0 obj <<
/Type /Font
/Subtype /Type1
/BaseFont /AGaramond-Semibold
/Encoding 25 0 R
/FontDescriptor 7 0 R
/FirstChar 0
/LastChar 255
/Widths 21 0 R
>> endobj
21 0 obj
[255 255 255 255 255 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 280 438 510 510 868 834
248 320 320 420 510 255 320 255 347 510 510 510 510
510 510 510 510 510 510 255 255 510 510 510 330 781
627 627 694 784 580 533 743 812 354 354 684 560 921
780 792 588 792 656 504 682 744 650 968 648 590 638
320 329 320 510 500 380 420 510 400 513 409 301 464
522 268 259 484 258 798 533 492 516 503 349 346 321
520 434 684 439 448 390 320 255 320 510 255 627 627
694 580 780 792 744 420 420 420 420 420 420 402 409
409 409 409 268 268 268 268 533 492 492 492 492 492
520 520 520 520 486 400 510 510 506 398 520 555 800
800 1044 360 380 549 846 792 713 510 549 549 510 522
494 713 823 549 274 354 387 768 615 496 330 280 510
549 510 549 612 421 421 1000 255 627 627 792 1016
730 500 1000 438 438 248 248 510 494 448 590 100 510
256 256 539 539 486 255 248 438 1174 627 580 627 580
580 354 354 354 354 792 792 790 792 744 744 744 268
380 380 380 380 380 380 380 380 380 380]
endobj
```

รูปที่ 10 ตัวอย่างแบบอักษรประเภทที่ 1 ในเอกสารพีดีเอฟ

ส่วนขยายแบบอักษรประเภทที่ 1 เป็นแบบอักษรที่เพิ่มเติมมาจากแบบอักษร ประเภทที่ 1 เพื่ออนุญาตในการสร้างรูปแบบตัวอักษรขนาดต่างๆ จากแบบอักษรแบบเดียว ในกรณีที่ไม่มีแบบอักษรนั้นในระบบที่กำลังอ่านเอกสารสามารถที่จะแสดงตัวอักษรแทนแบบอักษรนั้นได้

ตารางที่ 2 ข้อมูลส่วนต่างๆของแบบอักษรส่วนขยายประเภทที่ 1

ข้อมูล	ชนิดของข้อมูล	ระบุถึง
Subtype	ชื่อ	ชนิดของแบบอักษร ต้องระบุเป็น Multiple master Type1
BaseFont	ชื่อ	ชื่อของแบบอักษรโพสต์สคริปต์ที่ถูกกำหนดในแฟ้มโพสต์สคริปต์สำหรับใช้ในการแสดงข้อความ เป็นชื่อที่จะถูกใช้ในการโปรแกรมภาษาโพสต์สคริปต์
FirstChar	จำนวนเต็ม	รหัสตัวอักษรตัวแรกที่ใช้ในแถวลำดับที่กำหนดความกว้างของอักขระที่ใช้ในเอกสาร
LastChar	จำนวนเต็ม	รหัสตัวอักษรตัวสุดท้ายที่ใช้ในแถวลำดับที่กำหนดความกว้างของอักขระที่ใช้ในเอกสาร
Widths	แถวลำดับ	ขนาดของแต่ละตัวอักษรที่ใช้ในเอกสารและจะระบุลำดับตัวอักษรตัวแรกและตัวอักษณตัวสุดท้ายที่ใช้ในแบบอักษร
Encoding	พจนานุกรม	การเข้ารหัสตัวอักษรของแบบอักษรที่ใช้ในเอกสาร
FontDescriptor	พจนานุกรม	ข้อมูลที่เป็นรายละเอียดต่างๆที่จะใช่ในการแสดงตัวอักษร เช่น ชื่อแบบอักษร อักขระในแบบอักษรที่ถูกใช้ในการแสดง กรอบขนาดในการแสดงแบบอักษร

```
7 0 obj <<
/Type /Font
/Subtype /MMType1
/BaseFont /MinionMM_366_465_11
/FirstChar 32
/LastChar 255
/Widths 19 0 R
/Encoding 5 0 R
/FontDescriptor 6 0 R
>>
endobj
19 0 obj
[187 235 317 430 427 717 607 168 326 326 421
619 219 317 219 282 427 427 427 427 427 427
248 320 320 420 510 255 320 255 347 510 510 510 510
510 510 510 510 510 510 255 255 510 510 510 330 781
627 627 694 784 580 533 743 812 354 354 684 560 921
780 792 588 792 656 504 682 744 650 968 648 590 638
320 329 320 510 500 380 420 510 400 513 409 301 464
522 268 259 484 258 798 533 492 516 503 349 346 321
427 427 427 427 219 219 619 619
... omitted data ...
301 301 301 569 569 0 569 607 607 607 239 400
780 792 588 792 656 504 682 744 650 968 648 590 638
320 329 320 510 500 380 420 510 400 513 409 301 464
522 268 259 484 258 798 533 492 516 503 349 346 321
400 400 400 253 400 400 400 400 400]
endobj
```

รูปที่ 11 ตัวอย่างแบบอักษรส่วนขยายประเภทที่ 1 ในเอกสารพีดีเอฟ

แบบอักษรประเภทที่ 3 เป็นแบบอักษรที่แตกต่างจากแบบอักษรอื่นๆที่ใช้ในเอกสารพีดีเอฟ เนื่องจาก แบบอักษรประเภทที่ 3 กำหนดแบบอักษรด้วยตัวมันเอง ขณะที่พจนานุกรมของแบบอักษรอื่นๆเก็บข้อมูลพื้นฐานของแบบอักษรนั้นไว้ แบบอักษรประเภทที่ 3 มีความยืดหยุ่นมากกว่าแบบอักษรประเภทที่ 1 แต่มีคุณภาพในการแสดงด้อยกว่าโดยเฉพาะ ในการแสดงตัวอักษรขนาดเล็กบนอุปกรณ์แสดงผลที่มีรายละเอียดในการแสดงผลต่ำ

ตารางที่ 3 ข้อมูลส่วนต่างๆของแบบอักษรประเภทที่ 3

ข้อมูล	ชนิดของข้อมูล	ระบุถึง
Subtype	ชื่อ	ชนิดของแบบอักษร ต้องระบุเป็น ไทป์ทรี
FirstChar	จำนวนเต็ม	รหัสตัวอักษรตัวแรกที่ใช้ในแถวลำดับที่กำหนดความกว้างของอักขระที่ใช้ในเอกสาร
LastChar	จำนวนเต็ม	รหัสตัวอักษรตัวสุดท้ายที่ใช้ในแถวลำดับที่กำหนดความกว้างของอักขระที่ใช้ในเอกสาร
Widths	แถวลำดับ	ขนาดของแต่ละตัวอักษรที่ใช้ในเอกสารและจะระบุลำดับตัวอักษรตัวแรกและตัวอักษณตัวสุดท้ายที่ใช้ในแบบอักษร
CharProcs	พจนานุกรม	ข้อมูลแต่ละตัวในพจนานุกรมคือชื่อตัวอักษร และความสัมพันธ์ที่จะนำมาใช้ในการแสดงตัวอักษร
FontBBox	พจนานุกรม	กรอบสี่เหลี่ยมที่กำหนดขนาดของแบบอักษร
FontMatrix	พจนานุกรม	ข้อมูลที่จะนำมาใช้ในการกำหนดระยะตัวอักษร
Resource	พจนานุกรม	ทรัพยากรที่จะใช้ในการแสดงอักษร
Encoding	พจนานุกรม	การเข้ารหัสตัวอักษรของแบบอักษรที่ใช้ในเอกสาร

```
6 0 obj <<
/Type /Font
/Subtype /Type3
/Name /T36
/CharProcs 10 0 R
/Resources 8 0 R
/FontBBox [3 -241 875 856]
/FontMatrix [.001 0 0 .001 0 0]
/FirstChar 1
/LastChar 4
/Widths [47 44 49 37]
/Encoding 9 0 R
>> endobj
9 0 obj
<<
/Type /Encoding
/Differences [1/G01 /G02 /G03 /G04 ]
>>
endobj
10 0 obj
<<
/G01 11 0 R
/G02 12 0 R
/G03 13 0 R
/G04 14 0 R
>>
endobj
```

รูปที่ 12 ตัวอย่างชุดแบบตัวประเภทที่ 3 ในเอกสารพีดีเอฟ

แบบอักษรประเภททรูไทป์ เป็นแบบอักษรชนิดปรับขนาดได้ ปัจจุบันถูกนำมาใช้เป็นแบบอักษรมาตรฐานระบบปฏิบัติการวินโดวส์ และเป็นมาตรฐานหนึ่งในการแสดงตัวอักษรนี้ในเอกสารพีดีเอฟ ข้อกำหนดและรายละเอียดของแบบอักษรนี้ มีรูปแบบเหมือนกับแบบอักษรประเภทที่ 1

ตารางที่ 4 ข้อมูลส่วนต่างๆของแบบอักษรประเภททรูไทป์

ข้อมูล	ชนิดของข้อมูล	ระบุถึง
Subtype	ชื่อ	ชนิดของแบบอักษร ต้องระบุเป็น ทรูไทป์
BaseFont	ชื่อ	ชื่อของแบบอักษรโพสต์สคริปต์ที่ถูกกำหนดในแฟ้มโพสต์สคริปต์สำหรับใช้ในการแสดงข้อความ เป็นชื่อที่จะถูกใช้ในการโปรแกรมภาษาโพสต์สคริปต์
FirstChar	จำนวนเต็ม	รหัสตัวอักษรตัวแรกที่ใช้ในแถวลำดับที่กำหนดความกว้างของอักขระที่ใช้ในเอกสาร
LastChar	จำนวนเต็ม	รหัสตัวอักษรตัวสุดท้ายที่ใช้ในแถวลำดับที่กำหนดความกว้างของอักขระที่ใช้ในเอกสาร
Widths	แถวลำดับ	ขนาดของแต่ละตัวอักษรที่ใช้ในเอกสารและจะระบุลำดับตัวอักษรตัวแรกและตัวอักษณตัวสุดท้ายที่ใช้ในแบบอักษร
Encoding	พจนานุกรม	การเข้ารหัสตัวอักษรของแบบอักษรที่ใช้ในเอกสาร
FontDescriptor	พจนานุกรม	ข้อมูลที่เป็นรายละเอียดต่างๆที่จะใช้ในการแสดงตัวอักษร เช่น ชื่อแบบอักษร อักขระในแบบอักษรที่ถูกใช้ในการแสดง กรอบขนาดในการแสดงแบบอักษร

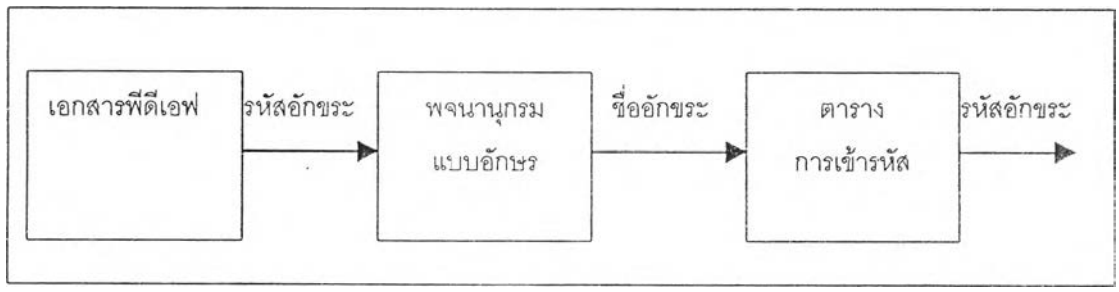
```
17 0 obj <<
/Type /Font
/Subtype /TrueType
/BaseFont /NewYork,Bold
/FirstChar 0
/LastChar 255
/Widths 23 0 R
/Encoding /WindowsEncoding
/FontDescriptor 7 0 R
>> endobj
23 0 obj
[0 333 333 333 333 333 333 333 0 333 333 333 333 333
333 333 333 333 333 333 333 333 333 333 333 333 333
333 333 0 333 333 333 303 500 666 666 882 848 303
446 446 507 666 303 378 303
... omitted data ...
303 530 1280 757 605 757 605 605 355 355 355 355 803
803 790 803 780 780 780 340 636 636 636 636 636 636
636 636 636 636]
endobj
7 0 obj
<<
/Type /FontDescriptor
/FontName /CordialUPC
/Flags 4
/FontBBox [ -250 -488 950 1000 ]
/ItalicAngle 0
/CapHeight 898
/Ascent 898
/MaxWidth 792
>>
endobj
```

รูปที่ 13 ตัวอย่างแบบอักษรประเภททรูไทป์ ในเอกสารพีดีเอฟ

2.1.4 การเข้ารหัสตัวอักษร[1]

การเข้ารหัสตัวอักษรเป็นส่วนสำคัญที่จะใช้ในการระบุตัวอักษร ว่าตัวอักษรนี้คือตัวอักษรอะไรมีรูปร่างอย่างไร เพื่อนำไปใช้ในการแสดงตัวอักษรและการค้นข้อความในเอกสารพีดีเอฟ การเข้ารหัสตัวอักษรในเอกสารพีดีเอฟใช้โครงสร้างการเข้ารหัสตัวอักษรที่มีความยืดหยุ่น โดยจะเก็บข้อมูลการเข้ารหัสและข้อมูลในการอธิบายโครงร่างตัวอักษรไว้แยกต่างหาก (ดังจะเห็นได้จากตัวอย่างแบบอักษรที่เคยได้กล่าวมาแล้ว ในหัวข้อที่ผ่านมา)

การแสดงข้อความในเอกสารโดยทั่วๆไป เอกสารต่างๆใช้รหัสอักขระในการระบุตัวอักษรในเอกสาร เช่น รหัสอักขระ 65 ก็จะคือตัวอักษร "A" หรือ รหัสอักขระ 161 ก็จะคือตัวอักษร "ก" แต่ในเอกสารพีดีเอฟ รหัสอักขระ65 ไม่จำเป็นต้องเป็นตัวอักษร "A" ถ้ารหัสอักขระ 65 ในพจนานุกรมแบบอักษร เข้ารหัส 65 /at รหัสอักขระ 65 ก็จะคือตัวอักษร "@" ถ้ารหัสอักขระ 161 ในพจนานุกรมแบบอักษร เข้ารหัส 161 /zero รหัสอักขระ 161 ก็จะคือตัวอักษร "0" การแสดงตัวอักษรหรือถอดรหัสตัวอักษรในเอกสารพีดีเอฟ จะอ่านค่ารหัสอักขระในเอกสารพีดีเอฟ ไปดูว่าอักขระนั้นมีชื่ออักขระว่าอะไร แล้วทำการแปลงรหัสอักขระนั้นตามตารางการเข้ารหัสที่ได้กำหนดไว้ (ดูรูปที่ 14 ประกอบ)



รูปที่ 14 ความสัมพันธ์ระหว่างรหัสตัวอักษรและชื่อตัวอักษรในเอกสารพีดีเอฟ

การเข้ารหัสตัวอักษรในเอกสารพีดีเอฟ มีอยู่ด้วยกัน 3 แบบ คือ

1. มาตรฐาน(Standard) คือการเข้ารหัสตัวอักษรจะเป็นไปตามมาตรฐานที่ข้อกำหนดเอกสารพีดีเอฟกำหนดไว้ เช่น การเข้ารหัสตัวอักษรตามมาตรฐานวินโดวส์ (WinAnsi) (ดูรูปที่ 15 ประกอบ) หรือ การเข้ารหัสตัวอักษรตามมาตรฐานอโดบี (Standard) (สามารถดูข้อกำหนดมาตรฐานการเข้ารหัสตัวอักษรตามข้อกำหนด ของเอกสารพีดีเอฟได้ที่ ภาคผนวก ก)

---

[1] Adobe Systems Incorporated, *PostScript Language Reference Manual, Third Edition*. Addison-Wesley, 1990, p.369

```
20 0 obj
<< /Type /Font
/Subtype /TrueType
/Name /F1
/BaseFont /CordiaUPC
/Encoding /WinAnsiEncoding
/FontDescriptor 22 0 R >>
endobj
```

รูปที่ 15 ตัวอย่างการเข้ารหัสตัวอักษรในเอกสารพีดีเอฟ
ตามข้อกำหนดมาตรฐานวินโดวส์

2. ผู้ใช้กำหนดการเข้ารหัสตัวอักษรเอง(Custom) คือ ข้อมูลการเข้ารหัสจะถูกกำหนดไว้ที่พจนานุกรมแบบอักษรแยกออกมาเป็นส่วนข้อมูลอีกส่วนหนึ่ง

```
4 0 obj
<<  /Name /F2
/Type /Font
/Subtype /Type3
/Encoding 9 0 R
/CharProcs 10 0 R
/Widths [47 44 49 37] >>
endobj
9 0 obj
<<
/Type /Encoding
/Differences [1/afii59681 /afii59682 /afii59684 /afii59687]
>>
endobj
```

รูปที่ 16 ตัวอย่างการเข้ารหัสตัวอักษรในเอกสารพีดีเอฟ
แบบผู้ใช้กำหนดการเข้ารหัสตัวอักษรเอง

3. แบบใช้ข้อกำหนดการเข้ารหัสตัวอักษรที่มีอยู่ในแฟ้มแบบอักษร(Built-In) คือ ไม่มีข้อมูลการเข้ารหัสตัวอักษรในเอกสารพีดีเอฟ ให้ใช้การเข้ารหัสตัวอักษรที่ถูกกำหนดอยู่ภายในแฟ้มแบบอักษรนั้น

```
14 0 obj
<<
/Type /Font
/Subtype /TrueType
/FirstChar 33
/LastChar 45
/Widths [ 400 376 417 313 405 347 0 417 0 382 347 243 405 ]
/BaseFont /BHDGMA+CordiaUPC+1
/FontDescriptor 11 0 R
/ToUnicode 12 0 R
>>
obj
```

รูปที่ 17 ตัวอย่างการเข้ารหัสตัวอักษรในเอกสารพีดีเอฟ
แบบใช้ข้อกำหนดที่มีอยู่ในแฟ้มแบบอักษร

2.2. มาตรฐานการเข้ารหัสอักขระไทย

2.2.1 มาตรฐานมอก.620

หน่วยงานมาตรฐานผลิตภัณฑ์อุตสาหกรรมไทยได้ประกาศใช้รหัสสำหรับอักษรไทยที่ใช้กับคอมพิวเตอร์เป็นครั้งแรก ที่มาตรฐานเลขที่ มอก.620-2529 ในราชกิจจานุเบกษา เล่ม 103 ตอนที่ 102 วันที่ 17 มิถุนายน พุทธศักราช 2529 แต่เนื่องจากยังขาดรายละเอียดเกี่ยวกับหมายเลขอักษรและชื่ออักษร ซึ่งเป็นส่วนสำคัญที่จะต้องใช้ในการจดทะเบียนอักษรไทยกับองค์การระหว่างประเทศว่าด้วยการมาตรฐาน หน่วยงานมาตรฐานผลิตภัณฑ์อุตสาหกรรมไทยจึงทำการแก้ไขและกำหนดมาตรฐานที่ มอก.620-2533 ดังนี้

อักขระในภาษาไทย ประกอบด้วย พยัญชนะ สระ วรรณยุกต์ ตัวเลขไทย และเครื่องหมายต่างๆ ดังนี้

พยัญชนะ ได้แก่

ก ข ฃ ค ฅ ฆ ง จ ฉ ช ฌ ญ ฎ ฏ ฐ ฑ ฒ ณ ด ต ถ ท ธ น บ ป ผ ฝ พ ฟ ภ ม ย ร ล ว ศ ษ ส ห ฬ อ ฮ

สระ ได้แก่

ะ า ำ ิ ี ึ ื ุ ู เ แ โ ใ ไ ฤ ฦ (นิคหิต) ◌ (พินทุ)

วรรณยุกต์และทัณฑฆาต ได้แก่

◌่ ◌้ ◌๊ ◌๋ และ ◌์ (ทัณฑฆาต)

ตัวเลขไทย ได้แก่

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙

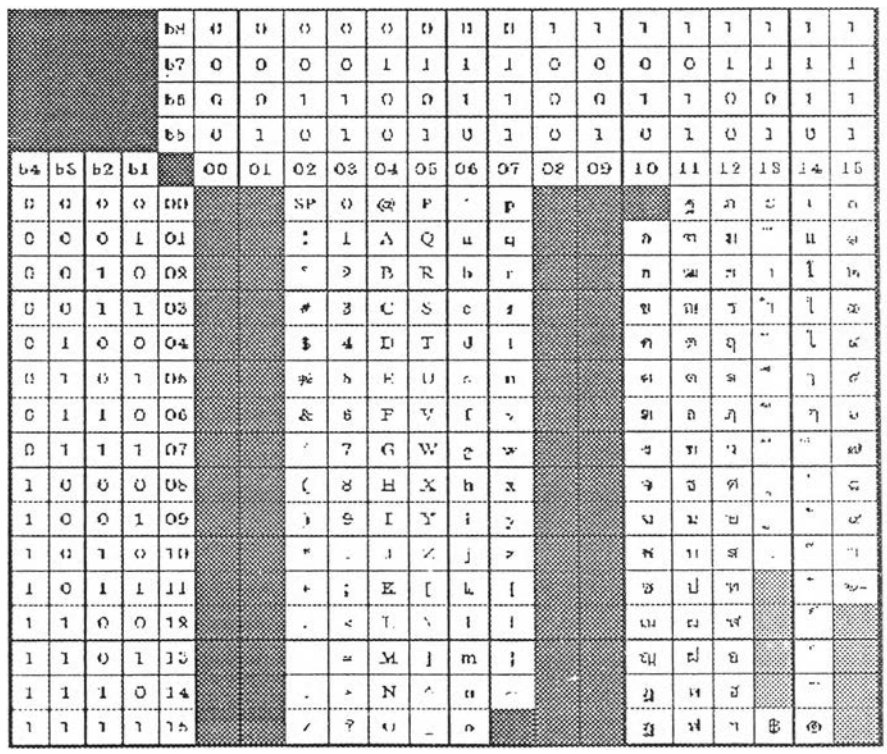
เครื่องหมายพิเศษ

เครื่องหมายพิเศษทั่วไป ได้แก่

ๆ ฯ ๚

เครื่องหมายพิเศษเฉพาะ ได้แก่

฿ (บาท) ◌็ (ยามักการ) ๏ (ฟองมัน) ฯๅ (อังคั่นคู่) และ ◌ (โคมูตร)



รูปที่ 18 ข้อกำหนดการเข้ารหัสตัวอักษรภาษาไทยตามมาตรฐาน มอก.620

2.2.2 มาตรฐานแบบอักษรไทยในระบบปฏิบัติการวินโดวส์

บริษัทไมโครซอฟต์ (Microsoft Inc.) ได้กำหนดการใช้และแสดงตัวอักษรภาษาไทยในระบบปฏิบัติการวินโดวส์ ตามข้อกำหนดมาตรฐาน มอก.620 และมาตรฐานยูนิโค้ด (Unicode) โดยกำหนดการเข้ารหัสที่การเข้ารหัสหน้า 874 มีรายละเอียดและข้อกำหนดดังนี้

ตารางที่ 5 ตารางการเข้ารหัสแบบอักษรไทยในระบบปฏิบัติการวินโดวส์

ตัวอักษร	รหัสยูนิโค้ด	ชื่อตัวอักษร
ก	0E01	afii59681
ข	0E02	afii59682
ฃ	0E03	afii59683
ค	0E04	afii59684
ฅ	0E05	afii59685
ฆ	0E06	afii59686
ง	0E07	afii59687
จ	0E08	afii59688
ฉ	0E09	afii59689
ช	0E0A	afii59690
ซ	0E0B	afii59691
ฌ	0E0C	afii59692
ญ	0E0D	afii59693
ฎ	0E0E	afii59694
ฏ	0E0F	afii59695
ฐ	0E10	afii59696
ฑ	0E11	afii59697
ฒ	0E12	afii59698
ณ	0E13	afii59699
ด	0E14	afii59700

ตัวอักษร	รหัสยูนิโค้ด	ชื่อตัวอักษร
ต	0E15	afii59701
ถ	0E16	afii59702
ท	0E17	afii59703
ธ	0E18	afii59704
น	0E19	afii59705
บ	0E1A	afii59706
ป	0E1B	afii59707
ผ	0E1C	afii59708
ฝ	0E1D	afii59709
พ	0E1E	afii59710
ฟ	0E1F	afii59711
ภ	0E20	afii59712
ม	0E21	afii59713
ย	0E22	afii59714
ร	0E23	afii59715
ฤ	0E24	afii59716
ล	0E25	afii59717
ฦ	0E26	afii59718
ว	0E27	afii59719
ศ	0E28	afii59720

ตารางที่ 5 (ต่อ) ตารางการเข้ารหัสแบบอักษรไทยในระบบปฏิบัติการวินโดวส์

ตัวอักษร	รหัสยูนิโค้ด	ชื่อตัวอักษร
ษ	0E29	afii59721
ส	0E2A	afii59722
ห	0E2B	afii59723
ฬ	0E2C	afii59724
อ	0E2D	afii59725
ฮ	0E2E	afii59726
ฯ	0E2F	afii59727
ะ	0E30	afii59728
ั	0E31	afii59729
า	0E32	afii59730
ำ	0E33	afii59731
ิ	0E34	afii59732
ี	0E35	afii59733
ึ	0E36	afii59734
ื	0E37	afii59735
ุ	0E38	afii59736
ู	0E39	afii59737
	0E3A	afii59738
฿	0E3F	afii59743
เ	0E40	afii59744
แ	0E41	afii59745
โ	0E42	afii59746
ใ	0E43	afii59747
ไ	0E44	afii59748
ๅ	0E45	afii59749
ๆ	0E46	afii59750
็	0E47	afii59751

ตัวอักษร	รหัสยูนิโค้ด	ชื่อตัวอักษร
่	0E48	afii59752
้	0E49	afii59753
๊	0E4A	afii59754
๋	0E4B	afii59755
์	0E4C	afii59756
ํ	0E4D	afii59758
๏	0E4F	afii59759
๐	0E50	afii59760
๑	0E51	afii59761
๒	0E52	afii59762
๓	0E53	afii59763
๔	0E54	afii59764
๕	0E55	afii59765
๖	0E56	afii59766
๗	0E57	afii59767
๘	0E58	afii59768
๙	0E59	afii59769
๚	0E5A	afii59770
๛	0E5B	afii59771
◌ิ	f701	f702
◌ี	f702	f702
◌ึ	f703	f703
◌ื	f704	f704
◌อ	f705	f705
◌้	f706	f706
◌๊	f707	f707
◌่	f708	f708

ตารางที่ 5 (ต่อ) ตารางการเข้ารหัสแบบอักษรไทยในระบบปฏิบัติการวินโดวส์

ตัวอักษร	รหัสยูนิโค้ด	ชื่อตัวอักษร
์	f709	f709
่	f70A	f70A
้	f70B	f70B
๊	f70C	f70C
๋	f70D	f70D
์	f70E	f70E
ญ	f70F	f70F
ฐ	f710	f710
ฑ	f711	f711

ตัวอักษร	รหัสยูนิโค้ด	ชื่อตัวอักษร
็	f712	f712
่	f713	f713
้	f714	f714
๊	f715	f715
๋	f716	f716
์	f717	f717
ฺ	f718	f718
ฺ	f719	f719
ฺ	f71A	f71A

2.3 การค้นข้อความโดยวิธีการเปรียบเทียบสายอักขระ[3]

การค้นข้อความโดยวิธีการเปรียบเทียบสายอักขระ หลักการของวิธีนี้คือ จะทำการเปรียบเทียบกลุ่มอักขระ 2 กลุ่ม คือกลุ่มอักขระของข้อความที่ต้องการค้นกับกลุ่มอักขระของข้อความที่ต้องทำการเปรียบเทียบทีละ 1 คู่อักขระจากซ้ายไปขวา การค้นโดยวิธีนี้เป็นการค้นหาจากข้อมูลจริงโดยตรง ไม่มีการสร้างรหัสขึ้นมาเป็นดัชนีเพื่อการเปรียบเทียบแต่อย่างใด การค้นหาข้อความเป็นการเปรียบเทียบที่ตรงกัน จะค้นพบข้อความได้ก็ต่อเมื่อมีคำที่ต้องการค้นปรากฏในข้อมูลเหล่านั้น เช่น ถ้าค้นคำว่า "Exact" จะค้นพบคำๆนี้ก็ต่อเมื่อมีคำว่า "Exact" ปรากฏอยู่จริงในกลุ่มอักขระนั้น

แนวคิดวิธีการค้นข้อความโดยวิธีการเปรียบเทียบสายอักขระ

2.3.1 การเปรียบเทียบสายอักขระตามแนวคิดของ Brute-force

เป็นแนวคิดเบื้องต้นในการทำการเปรียบเทียบข้อความอย่างตรงไปตรงมา คือ เปรียบเทียบกันทีละตัวอักษร ถ้าตัวอักษรตัวแรกของข้อความที่ต้องการค้นตรงกับตัวอักษรของข้อความที่นำมาเปรียบเทียบก็จะขยับไป 1 ตัวอักษรแล้วเปรียบเทียบตัวอักษรถัดไปเรื่อยๆ ถ้าไม่ตรงกันเมื่อไรจะเลื่อนตัวอักษรของข้อความที่ต้องการค้นกลับมาที่ตัวอักษรตัวแรก วิธีการนี้จะเสียเวลามากเพราะจะต้องเปรียบเทียบอักขระทุกอักขระของข้อความที่นำมาเปรียบเทียบ

[3] Jun-ichi Aoe, *Computer Algorithm: string pattern matching strategies*, IEEE Computer Society Press, 1994,p.1-4

2.3.2 การเปรียบเทียบสายอักขระตามแนวคิดของ Knuth-Morris-Pratt

เป็นแนวคิดที่พัฒนาต่อมาจาก Brute-force โดยแก้ไขส่วนที่บกพร่องในเรื่องของการขยับตัวอักษรเพื่อนำไปเปรียบเทียบ โดยในการขยับตัวอักษรจะขยับได้มากกว่าครั้งละ 1 ตัวอักษรแต่จะไม่เกินความยาวของคำที่ค้น การที่ขยับได้ครั้งละมากกว่า 1 ตัวอักษรทำให้การค้นไม่ต้องทำการเปรียบเทียบทุกตัวอักษรทำให้ค้นข้อความได้รวดเร็วกว่าวิธีการของ Brute-force

2.3.3 การเปรียบเทียบสายอักขระตามแนวคิดของ Boyer และ Moore

เป็นแนวคิดที่ได้รับความนิยม มีแนวคิดคล้ายกับ 2 แบบแรกที่ได้กล่าวมา แต่เพิ่มประสิทธิภาพในการเลื่อนตัวอักษรให้ดีขึ้นกว่าทั้ง 2 แบบ โดยมีแนวความคิดดังนี้

ให้ i คือ ตำแหน่งตัวอักษรของข้อความที่ต้องการค้น (Pattern)

j คือ ตำแหน่งตัวอักษรของข้อความทั้งหมด (String)

n คือ จำนวนตัวอักษรของข้อความที่ต้องการค้น (Length of Pattern)

ในการเลื่อนตัวอักษรเปรียบเทียบ จะมีวิธีการเลื่อนตัวอักษรดังนี้

ให้ j เริ่มต้นเท่ากับ n

ถ้า String[j] ตรงกับตัวอักษรตัวใดๆใน Pattern[i] ให้ j = j + (n - i)

ถ้าไม่แล้ว ให้ j = j + n

ตัวอย่างเช่น *Pattern = "corn"* *String = "Oaks of acorns"*

ขั้นตอนที่ 1

```
c o r n
O a k s  o f  a c o r n s   => j = 4,  String[4] ไม่ตรงกับ 'c','o','r','n'
```

ขั้นตอนที่ 2 *เมื่อ String[4] ไม่ตรงกับตัวอักษรใดๆ ให้ j = j + n => j = 4 + 4*

```
    c o r n
O a k s  o f  a c o r n s   => j = 8,  String[8] ไม่ตรงกับ 'c','o','r','n'
```

ขั้นตอนที่ 3 *เมื่อ String[8] ไม่ตรงกับตัวอักษรใดๆ ให้ j = j + n => j = 8 + 4*

```
        c o r n
O a k s  o f  a c o r n s   => j = 12, String[12] ตรงกับ Pattern[3]
```

ขั้นตอนที่ 4 *เมื่อ String[12] ตรงกับ Pattern[3] ให้ j = j + (n-i) => j = 12 + 1*

```
      c o r n
O a k s  o f  a c o r n s   => j = 13, พบข้อความที่ต้องการค้น
```

จากตัวอย่างจะเห็นว่าพบ "corn" ในข้อความ "Oaks of acorns" แต่ถ้าหากเปรียบเทียบแล้ว ข้อความที่ต้องการค้นยังไม่ตรงกับข้อความที่ใช้เปรียบเทียบ ก็ให้ทำการเลื่อนตัวอักษรและตรวจสอบเหมือนที่ได้กล่าวมาแล้ว