การใช้ sgml-tools กับภาษาไทย

พูลลาภ วีระธนาบุตร $Id: sgmltools.sgml,v 1.1 1998/09/28 11:26:44 poon-v Exp poon-v $

เอกสารฉบับจะอธิบายการใช้ภาษาไทยกับ SGML-tools และการเขียนเอกสาร SGML เบื้องต้นเพื่อใช้กับ SGML-tools. ผู้อ่านสามารถติดตามเอกสารฉบับล่าสุดได้ที่ http://www.fedu.uec.ac.jp/ZzzThai/Linux

Contents

1 แนะนำ SGML-tools

SGML-tools เป็นชุดโปรแกรมสำหรับทำเอกสารโดยการเขียนเอกสารในรูปแบบของ SGML (Standard Generalized Markup Language) และใช้โปรแกรมจาก SGML-tools แปลงเป็นเอกสารแบบอื่นๆได้แก่ เอกสารแบบ HTML, lyx, latex, dvi, Postscript, plain text, rich text และ GNU info.

SGML-tools เดิมชื่อ Linuxdoc-sgml เขียนขึ้นเพื่อใช้ใน *Linux Document Project (LDP)* `<http://sunsite.unc.edu/linux>`. ทำให้การเขียนเอกสารสำหรับโปรเจคนี้มีรูปแบบมาตรฐานที่แน่นอน. ตามหลักของ LDP แล้วเอกสาร HOWTO ต้องเขียนในรูปแบบ SGML, ส่วนเอกสาร mini-HOWTO จะเขียนด้วยรูปแบบ HTML. license ของ LDP สามารถดูได้จากเอกสาร *Linux Document Project copying License*

<http://sunsite.unc.edu/linux/LDP-COPYRIGHT.html>. ซึ่งเอกสารเหล่านี้มี copyright โดยคนที่เขียน และไม่ใช่ Public Domain.

ผู้อ่านสามารถ download SGML-tools เวอร์ชั่นล่าสุดได้ที่ *SGML tools download* <http://www.sgmltools.org/download.html>. โดยทั่วไปแล้ว SGML-tools มักมาพร้อมๆกับ Linux. เนื่องจาก SGML-tools เป็นชุดโปรแกรมที่แปลงเอกสารเป็นเอกสารรูปแบบอื่นๆ เพราะฉะนั้นผู้ใช้ควรจะมีโปรแกรมที่จัดการเอกสารรูปแบบนั้นๆไว้ใช้ด้วย ได้แก่ `groff`, `TeX`, `LaTeX`, `flex`, `gawk` และ `lyx`.

ในเอกสารฉบับนี้จะอธิบายการเขียนเอกสารภาษาไทยในแบบของ SGML (linuxdoc) เบื้องต้น เพื่อให้ผู้สนใจการเขียนเอกสาร HOWTO สำหรับ Linux เขียนเอกสารอย่างมีประสิทธิภาพและรูปแบบมาตรฐานเดียวกัน. และจะอธิบายการใช้ภาษาไทยกับการแปลงเอกสารที่แปลงเป็นแบบ HTML, LaTeX, และ text เท่านั้น.

2 การเขียนเอกสารในรูปแบบ SGML (linuxdoc)

หลักการเขียนเอกสารแบบ SGML คล้าย HTML มาก กล่าวคือใช้ tags ในการกำหนดรูปแบบของเอกสาร.

2.1 โครงสร้างของเอกสาร SGML-tools

โครงสร้างเอกสาร SGML สำหรับ SGML-tools มีรูปแบบทั่วไปดังนี้

```
<!doctype linuxdoc system>

<article>

<title> DOCUMENT TITLE
<author> AUTHOR NAME
<date> DATE
<abstract>
ABSTRACT HERE
</abstract>

<toc> <!-- table of content -->

<sect> SECTION
<p>
CONTENT .......
</article>
```

เอกสารที่ใช้ SGML-tools ต้องมีคำสั่ง <`!doctype linuxdoc system`> ทุกครั้ง. เพื่อเป็นการบอกให้ SGML-tools รู้ว่าคำสั่ง(TAG)ในเอกสารนี้ถูกกำหนดโดย linuxdoc. รายละเอียดทางเทคนิคของ SGML สามารถอ่านได้จาก guide.sgml ซึ่งเป็นเอกสารแนะนำการใช้ SGML-tools ที่มากับตัวโปรแกรม.

2.2 การใช้ TAG

การใช้ TAG ในไฟล์ SGML คล้ายๆกับ TAG ของ HTML คือมีตัวเปิดและปิด. ไวยกรณ์ทั่วคือ

```
<TAG> phrase </TAG>
```

คำสั่งข้างบนสามารถย่อสั้นๆได้ดังนี้

```
<TAG/ phrase /
```

ในกรณีหลังเหมาะสำหรับการเขียนคำสั้นๆ และไม่เหมาะสำหรับการเขียนชื่อไฟล์เต็มๆของ UNIX เพราะมีเครื่องหมาย / อยู่ซึ่งถือเป็นส่วนหนึ่งของคำสั่ง.

TAG บางชนิดไม่จำเป็นต้องมีตัวปิดก็ได้เช่น <P> ซึ่งหมายถึงคำสั่งกำกับย่อหน้าเป็นต้น. อีกวิธีหนึ่งที่ใช้สำหรับการย่อหน้าคือ การเว้นบรรทัดว่างหนึ่งบรรทัด.

2.3 เครื่องหมายพิเศษ

การเขียนเครื่องหมายพิเศษมีความจำเป็นเมื่อผู้ใช้ต้องการเขียนตัวอักษรที่เป็นคำสั่งเช่น "<ภาษาไทย>" เพื่อไม่ให้แปลเครื่องหมาย <> เป็น TAG ผู้ใช้สามารถเขียนตัวอย่างที่แสดงได้โดย `<ภาษาไทย>` เป็นต้น. เครื่องหมายพิเศษที่ใช้บ่อยได้แก่,

- เขียน `&` แทนเครื่องหมาย ampersand (&),
- เขียน `<` แทนเครื่องหมาย left bracket (<),
- เขียน `>` แทนเครื่องหมาย right bracket (>),
- เขียน `&etago;` แทนเครื่องหมาย left bracket with a slash (</)
- เขียน `$` แทนเครื่องหมาย dollar sign ($),
- เขียน `#` แทนเครื่องหมาย hash (#),
- เขียน `%` แทนเครื่องหมาย percent (%),
- เขียน `˜` แทนเครื่องหมาย tilde (˜),
- เขียน `&dquot;` แทนเครื่องหมาย double quot ".

เครื่องหมายพิเศษอื่นๆที่ใช้ได้กับ SGML-tools 1.0.x สามารถดูได้จาก `guide.sgml`.

2.4 การเขียนข้อความแบบ verbatim

เมื่อผู้ใช้ต้องการแสดงข้อความที่ไม่ต้องการให้แปลตามภาษา SGML ให้เขียนข้อความเหล่านั้นใน environment ที่ชื่อ verbatim ดังนี้

```
<verb>
This is the verbatim environment.
<> & ; ~ You can write special characters here.
</verb>
```

ตัวอย่างข้างจะแสดงผลเป็นดังนี้

```
This is the verbatim environment.
<> & ; ~ You can write special characters here.
```

การใช้ <verb;>...<verb>/ ยังไม่เหมาะกับการใช้ภาษาไทยใน SGML มากนักเพราะขณะนี้ยังไม่สามารถแสดงผลภาษาไทยใน verbatim environment ของ LaTeX ได้. ใน verbatim environment มีข้อกำหนดการเขียนอักษรบางตัวดังนี้

- ให้เขียน `&ero;` เพื่อแสดงตัวอักษร ampersand,
- ให้เขียน `&etago;` เพื่อแสดงตัวอักษร </,
- ห้ามเขียน `\end{verbatim}` ใน `verb` environment, ซึ่งวลีนี้เป็นคำบอกเลิก `verbatim` environment ของ LaTeX.

นอกจาก `verb` environment แล้วยังมี environment อื่นที่คล้ายๆกันคือ `code`. ตัวอย่างเช่น

```
<code>
This is the code environment.
</code>
```

การแสดงผล :

```
This is the code environment.
```

`code` environment จะแตกต่างกับ `verb` environment ตรงที่มีเส้นกั้นข้อความ(horizontal rule)ให้.

ผู้ใช้ยังสามารถใช้ `tscreen` environment ประกอบกับ `verb` หรือ `code` เพื่อล่นข้อความ(indent)ไปทางขวา. ตัวอย่างเช่น,

```
<tscreen><verb>
Using verb environmment with tscreen environment.
</verb></tscreen>
```

การแสดงผล:

```
    Using verb environmment with tscreen environment.
```

`quote` environment คล้าย `tscreen` environment มากแต่แตกต่างกันตรงที่ไม่ใช้ฟอนต์ `tt` (fixed width font) ในการแสดงผล. ตัวอย่างเช่น,

```
<quote>
This is the quote environment.
</quote>
```

การแสดงผล:

> This is the quote environment.

2.5 การเขียนรายการ (list)

การเขียนรายการ(list) สามารถเขียนได้ 3 แบบคือ

- `itemize`, เป็นรายการที่ไม่ต้องการตัวเลขกำกับ
- `enum`, เป็นรายการที่กำกับตัวเลข
- `descrip`, เป็นการยกหัวข้อและอธิบายประกอบ

ตัวอย่างการเขียนรายการแบบ `itemize`

```
<itemize>
<item>รายการที่หนึ่ง
<item>รายการต่อไป
<itemize>/
```

การแสดงผล:

- รายการที่หนึ่ง
- รายการต่อไป

ตัวอย่างการเขียนรายการแบบ `enum`,

```
<enum>
<item>รายการที่หนึ่ง
<item>รายการต่อไป
<enum>/
```

การแสดงผล:

1. รายการที่หนึ่ง
2. รายการต่อไป

การเขียนรายการแบบ `descrip` จะแตกต่างจากตัวอย่างอื่นเล็กน้อยดังนี้

```
<descrip>
<tag/หัวข้อที่หนึ่ง/ คำอธิบาย
<tag/หัวข้อต่อไป/ คำอธิบาย
<descrip>/
```

การแสดงผล:

หัวข้อที่หนึ่ง
    คำอธิบาย

หัวข้อต่อไป
    คำอธิบาย

2.6 สรุปที่ใช้บ่อย TAG อื่นๆ

<article> ... </article>

เป็นคำสั่งบอกเริ่มเอกสาร ซึ่งขณะนี้เข้าใจว่าใช้ได้เฉพาะ article เท่านั้น ข้อความที่อยู่ระหว่างนี้ถือเป็นเนื้อหาของเอกสาร และคำสั่งที่อยู่ก่อนหน้านี้ถือเป็น preamble.

<author>

ใช้ระบุชื่อผู้เขียนเอกสาร

<date>

ใช้ระบุวันที่ที่เขียนเอกสาร

<abstract> ... </abstract>

เป็นส่วนสำหรับเขียนบทคัดย่อ อธิบายเนื้อหาของเอกสาร

<toc>

คำสั่งสร้างสารบัญ

<p>

ใช้ในการย่อหน้า โดยปรกติแล้ว TAG นี้ควรตามหลัง `<sect>` เพื่อเป็นการบอกว่าพร้อมที่จะเขียนเนื้อหาหลังจากประกาศหัวข้อ. การย่อหน้าอีกวิธีหนึ่งสามารถทำได้โดยการเว้นบรรทัดว่างหนึ่งบรรทัด ซึ่งจะนิยมมากกว่าใช้ <p>. TAG นี้ไม่จำเป็นต้องมีตัวปิดท้าย.

<sect>

เป็นคำย่อของ section ใช้สำหรับกำกับหัวข้อ ในกรณีนี้จะแสดงตัวเลขหัวข้อเป็น 1, 2, 3, ... ตามลำดับ

<sect1>

คำสั่งประกาศหัวข้อย่อย ใช้ประกาศภายในหัวข้อหลัก. ในกรณีจะมีตัวเลขเป็น 1.1, 1.2, 1.3, ... เป็นต้น. หัวข้อย่อยถัดจากนี้คือ `<sect2>`, `<sect3>` และ `<sect4>`

<tt> ... </tt>

เป็นคำสั่งเปลี่ยนฟอนต์เป็นฟอนต์พิมพ์ดีด(Type writer). มักใช้ในการเขียนชื่อคำสั่ง เป็นต้น

<url url="URL" name="NAME">

ใช้ใส่ URL ในเนื้อหา. ถ้าแปลงเป็น HTML format, NAME จะลิงค์ไปยัง URL นั้นๆ. ถ้าเป็น text หรือ LaTeX, จะแสดงด้วยชื่อแล้วตามด้วยที่อยู่ URL.

<htmlurl url="HTML_URL" name="NAME">

ใช้เขียน URL ของ HTML เช่น `<htmlurl url="mailto:poon-v@fedu.uec.ac.jp" name="`พูลลาภ วีระธนาบุตร" เป็นต้น

<label id="ID">

ใช้ในการเลือกตำแหน่งอ้างอิงของเอกสาร. คำสั่งนี้มักใช้ควบคู่กับกับคำสั่ง `<ref id="ID" name="NAME">`

<ref id="ID" name="NAME">

เป็นคำสั่งบอกตำแหน่งอ้างอิงในเอกสาร. ถ้าเป็น HTML format, จะเป็นการลิงค์กันภายในเอกสาร. ถ้าเป็น LaTeX format จะเป็นการบอกตำแหน่งหน้าที่อ้างอิง เป็นต้น.

ผู้ใช้สามารถอ่านวิธีการใช้คำสั่งอื่นๆได้จาก `guide.sgml` ที่มากับ SGML-tools.

ผู้ใช้สามารถเขียนภาษาไทยในไฟล์ SGML ได้แต่ไม่ควรเขียนภาษาไทยใน `verb` หรือ `code` environment ซึ่งต้องหาทางแก้ไขกันต่อไป.

3 การใช้ SGML-tools

ผู้ใช้สามารถเช็คไวยกรณ์ที่เขียนในไฟล์ SGML ได้โดย

```
% sgmlcheck foo.sgml
```

ในที่นี้จะขอกล่าวถึงการแปลงเอกสารภาษาไทยแบบ SGML ให้เป็นเอกสารแบบอื่นๆ เฉพาะแบบ Plain Text, HTML และ LaTeX เท่านั้น.

3.1 การผลิตเอกสารภาษาไทยแบบ Plain Text

ผู้ใช้สามารถแปลงเอกสารภาษาไทยแบบ SGML ที่เขียนไว้เป็นแบบ Plain Text ได้ด้วยคำสั่ง

```
% sgml2txt -c latin foo.sgml
```

ถ้าไม่เกิดข้อผิดพลาดใดๆ ผู้ใช้จะได้ไฟล์ใหม่ที่ชื่อ `foo.txt` ในไดเรกทอรี่ที่ทำงานอยู่.

3.2 การผลิตเอกสารภาษาไทยแบบ HTML

ผู้ใช้สามารถแปลงเอกสารภาษาไทยแบบ SGML ที่เขียนไว้เป็นแบบ HTML ได้ด้วยคำสั่ง

```
% sgml2html foo.sgml
```

ถ้าไม่เกิดข้อผิดพลาดใดๆ ผู้ใช้จะได้ไฟล์ใหม่ที่ชื่อ `foo.html` และ `foo-1.html, foo-2.html, ...` ในไดเรกทอรี่ที่ทำงานอยู่. ซึ่ง SGML-tools จะแบ่งเป็นไฟล์ย่อยๆตามหัวข้อ(section)ที่ระบุไว้. ในที่นี้ผู้ใช้ไม่ต้องใส่ option `-c latin`, เอกสารที่ผลิตออกมาจะเป็นรหัส 8-bit ซึ่งตรงกับรหัส tis-620 ภาษาไทยที่ใช้อยู่บนอินเตอร์เน็ตทั่วไป. เพื่อความสะดวกในการใช้งานยิ่งขึ้น, ผู้เขียนได้เขียน Perl script สำหรับแปลงเอกสาร HTML ที่เขียนในรูปของ `&LATIN_CHAR_NAME;` ให้เป็นรหัส tis-620 และมี option สั่งให้เรียก `cttex` มาตัดคำเพื่อความสวยงามด้วย.

```
#!/usr/bin/perl

################################################################
# sgmlthtml:
# Run sgml2html and convert latin symbols to Thai characters.
# This program will change, for example, a to "\340".
#
# by Poonlap Veeratanabutr <poon-v@fedu.uec.ac.jp>
# $Id: sgmlthtml,v 1.1 1998/09/28 06:51:52 poon-v Exp poon-v $
################################################################
```

```
sub error {
   print STDERR "usage: $0 [-c] file.sgml\n";
   print STDERR "     -c , use cttex to fill <WBR>\n";
   exit;
}

# command line processing.
if( $#ARGV == -1 ){
   &error;
} elsif ( $#ARGV == 0 && $ARGV[0] ne "-c"){
   $file = $ARGV[0];
   $cut = 0;
} elsif ( $#ARGV == 1 ){
   if( $ARGV[0] eq "-c"){
      $cut = 1;
      $file = $ARGV[1];
   } elsif( $ARGV[1] eq "-c"){
      $cut = 1;
      $file = $ARGV[0];
   } else {
      &error;
   }
} else {
   &error;
}

# run sgml2html
if( system( "sgml2html $file" ) != 0 ){
   exit;
}

# lookup table for what to change and not to change
%lookuptbl = (quot, "quot",amp,"amp","lt","lt","gt","gt",copy,"\251",reg,"\256",
          micro,"\265",Agrave,"\300",Aacute,"\301",Acirc,"\302",Atilde,"\303",
          Auml,"\304",Aring,"\305",AElig,"\306",Ccedil,"\307",Egrave,"\310",
          Eacute,"\311",Ecirc,"\312",Euml,"\313",Igrave,"\314",Iacute,"\315",
          Icirc,"\316",Iduml,"\317",ETH,"\320",eth,"\320",Ntilde,"\321",
          Ograve,"\322",Oacute,"\323",Ocirc,"\324",Otilde,"\325",Ouml,"\326",
          Oslash,"\330",Ugrave,"\331",Uacute,"\332",THORN,"THORN{\thai }",Thorn,"Thron",
          szlig,"\337",agrave,"\340",aacute,"\341",acirc,"\342",atilde,"\343",
          auml,"\344",aring,"\345",aelig,"\346",ccedil,"\347",egrave,"\350",
          eacute,"\351",ecirc,"\352",euml,"\353",igrave,"\354",iacute,"\355",
          icirc,"\356",iuml,"\357",eth,"\360",ntilde,"\361",ograve,"\362",
          oacute,"\363",ocirc,"\364",otilde,"\365",ouml,"\366",oslash,"\370",
          ugrave,"\371",uacute,"\372",ucirc,"\373",uuml,"uuml",yacute,"yacute",
          yuml,"yuml");
```

```
# separate directory and file name from the given file name
if( $file =~ /(.*)[\/](.+)$/ ){
   $dir = $1;
   $file = $2;
}
$file =~ s/(.+)\.sgml/$1/; # get rootname

if( length( $dir ) == 0 ){
   opendir( DIR, ".");
} else {
   opendir( DIR, "$dir" );
}
@html = grep {/$file.*\.html$/} readdir( DIR ); # get all related html files
closedir( DIR );

foreach $html_file (@html) {
   print "Processing file $html_file\n";
   open( INPUT, "$html_file");
   open( OUTPUT,">$html_file.tmp" );

   while( <INPUT> ){
      $line = $_;
      $beg = index( $line, "&");
      $end = index( $line, ";");
      while( $beg >= 0  && $end > $beg  ){
         $target = substr( $line, $beg+1, $end-$beg-1);
         $thai = $lookuptbl{ $target };
         if( $target eq $thai ){
            $line = sprintf( "%s%s%s", substr( $line, 0, $beg), '&' . $thai . ';',
                        substr( $line, $end+1 ));

         } else {
            $line = sprintf( "%s%s%s", substr( $line, 0, $beg), $thai,
                        substr( $line, $end+1 ));
         }
         $beg = index( $line, "&", $beg+1);
         $end = index( $line, ";", $beg);
      }
      print OUTPUT "$line";
   }
   close( INPUT );
   close( OUTPUT );
   if( $cut == 1 ){
      system( "cttex 0 < $html_file.tmp > $html_file" );
      system( "rm -f $html_file.tmp" );
   } else {
      rename( "$html_file.tmp", "$html_file");
```

```
    }
}

# EOF
```

ผู้ใช้สามารถเรียกคำสั่ง `sgmlthtml` แทน `sgml2html` สำหรับเอกสาร SGML ที่มีภาษาไทยได้ดังนี้

```
% sgmlthtml foo.sgml -c
```

จากตัวอย่างดังกล่าว, `sgmlthtml` จะเรียกคำสั่ง `sgml2html` ซึ่งจะผลิตไฟล์แบบ HTML หลายไฟล์. จากนั้นโปรแกรม `sgmlthtml` จะพยายามเปลี่ยน `&LATIN_CHAR_NAME;` ให้เป็นรหัส tis-620 (ถ้ามี, โดยปรกติ `sgml2html` จะผลิตไฟล์ที่เผอิญตรงตาม tis-620 อยู่แล้ว) และเรียก `cttex` มาตัดคำ(ทุกไฟล์ที่เกี่ยวข้องกับ `foo.sgml`) ให้การแสดงผลบน browser สวยงามขึ้น. การที่จะใช้คำสั่งนี้, ผู้ใช้ต้องมีโปรแกรม `cttex` อยู่ในระบบด้วย. หากไม่ต้องการให้ `cttex` ตัดคำ, ไม่ต้องใส่ option `-c` ดังตัวอย่าง

```
% sgmlthtml foo.sgml
```

3.3 การผลิตเอกสารภาษาไทยแบบ LaTeX

ผู้ใช้ต้องมีชุดโปรแกรม LaTeX และติดตั้งภาษาไทยเรียบร้อยแล้วจึงจะสามารถแปลงเอกสารจาก SGML เป็น LaTeX ได้. การติดตั้งภาษาไทยใน LaTeX สามารถอ่านได้จากเอกสาร "การใช้ภาษาไทยกับ *LaTeX* `<http://www.fedu.uec.ac.jp/ZzzThai/Linux/thailatex/thailatex>`" .

ผู้ใช้สามารถผลิตเอกสาร LaTeX(`.tex`) ที่มีภาษาไทยจากเอกสารแบบ SGML ได้โดยคำสั่ง

```
% sgml2latex --output=tex foo.sgml
```

จากคำสั่งข้างบน `sgml2latex` จะสร้างไฟล์ `foo.tex` ในไดเรกทอรี่ที่ทำงานอยู่. ผู้ใช้ยังไม่สามารถนำไฟล์นี้ไปใช้กับ LaTeX ได้โดยตรง, จะต้องแก้ส่วน preamble ของไฟล์นี้เล็กน้อยก่อนนำไปใช้กับ `cttex` และ `latex` ต่อไป.

ผู้ใช้ต้องลบบรรทัดต่อไปนี้ออกจาก preamble ของไฟล์ `foo.tex`

```
\usepackage[latin1]{inputenc}
\usepackage{t1enc}
\usepackage{babel}
```

และใส่บรรทัดต่อไปนี้แทนลงไปแทน

```
\usepackage{thai}
```

จากนั้นจึงสั่งคำสั่ง `cttex` และ `latex` ตามลำดับดังตัวอย่าง

```
% cttex < foo.tex > foo_ok.tex
% latex foo_ok.tex
```

จะเห็นได้ว่าเนื่องจาก SGML-tools ไม่ได้ผลิตมาเพื่อใช้กับภาษาไทยโดยเฉพาะจึงใช้ยุ่งยาก. ผู้เขียนพยายามลดขั้นตอนเหล่านี้โดยเขียน Perl script ชื่อ `sgmltlatex` ทำหน้าที่ต่างๆเหล่านี้โดยอัตโนมัติ.

```
#!/usr/bin/perl

###############################################################################
# sgmltlatex:
# Processing SGML or LaTeX file for Thai language.
#
#
# by Poonlap Veeratanabutr <poon-v@fedu.uec.ac.jp>
# $Id: sgmltools.sgml,v 1.1 1998/09/28 11:26:44 poon-v Exp poon-v $
###############################################################################
sub error {
   print STDERR "$0 - Processing SGML or LaTeX file for Thai language.\n";
   print STDERR "usage: $0 [-[t|d|p] sgml_file] [-l latex_file] \n";
   print STDERR "     -t , create latex_file file from sgml_file\n";
   print STDERR "     -d , create dvi file from sgml_file and leave latex_file\n";
   print STDERR "     -p , create Postscript file and leave latex_file\n";
   print STDERR "     -l , run latex on latex_file\n";
   exit( -1 );
}

# command line processing
if( $#ARGV != 1 ){
   &error;
} elsif ( $ARGV[0] eq "-l" ){
   $option = "l";
} elsif ( $ARGV[0] eq "-t" ){
   $option = "t";
} elsif ( $ARGV[0] eq "-d" ){
   $option = "d";
} elsif ( $ARGV[0] eq "-p" ){
   $option = "p";
} else {
   &error;
}
$file = $ARGV[1];
$tempfile = "$file" . ".tmp";

if( $option ne "l" ){
   if( system( "sgml2latex --output=tex $file" ) != 0 ){
      exit( -1 );
   }

# separate directory and file name from the given file name
   if( $file =~ /(.*)[\/](.+)$/ ){
      $dir = $1;
      $file = $2;
   }
```

```
    if( $file =~ m/(.+)\.sgml$/ ){
       $rootname = "$1";
    } else {
       $rootname = $file;
    }
    $file = $rootname . ".tex";

    print "Processing file $file\n";

    open( INPUT, "$file" );
    open( OUTPUT, ">" . "$tempfile" );
    $preamble = 1;
    while( <INPUT> ){
       if( $preamble == 1 ){
          if( /^\\begin\s*{\s*document\s*}\s*$/ ){
             print OUTPUT "\\usepackage{thai}\n";
             $preamble = 0;
             break;
          }
          s/^\\usepackage.*{\s*inputenc.*}\n//;
          s/^\\usepackage.*{\s*babel.*}\n//;
          s/^\\usepackage.*{\s*t1enc.*}\n//;
          print OUTPUT;

       } else {
          print OUTPUT;
       }
    }
    close( INPUT );
    close( OUTPUT );


 } else {
    unless( $file =~ m/(.+)\.tex$/ ){
       print STDERR "$0 needs .tex file\n";
       &error;
    }
    rename( "$file", "$tempfile" );
 }


 if( $option eq "t" ){
    rename( "$tempfile", "$file" );
    exit( 0 );
 } else {
    system( "cttex < $tempfile > $file" );
    system( "latex $file" );
```

```
   rename( "$tempfile", "$file" );
}

if( $option eq "p" ){
   $psfile = "$rootname" . ".ps";
   $dvifile = "$rootname" . ".dvi";
   system( "dvips -o $psfile $dvifile" );
}

exit( 0 );

#EOF
```

ผู้ใช้สามารถเรียกคำสั่ง `sgmltlatex` แทน `sgml2latex` สำหรับเอกสาร SGML ที่มีภาษาไทย. ถ้าต้องการผลิตเอกสารแบบ `.dvi`,

```
% sgmltlatex -d foo.sgml
```

หากไม่มีข้อผิดพลาดเกิดขึ้น, จะได้ `foo.dvi` ในไดเรกทอรี่ที่ทำงานอยู่. ถ้าต้องการผลิตไฟล์แบบ Postscript ให้สั่งคำสั่ง

```
% sgmltlatex -p foo.sgml
```

หรือถ้าต้องการผลิตเอกสารแบบ LaTeX (`.tex`) ให้ใช้คำสั่ง

```
% sgmltlatex -t foo.sgml
```

ในกรณีที่ใช้ option `"d"` หรือ `"p"` หรือ `"t"` จะได้ไฟล์แบบ LaTeX (`.tex`) ด้วย. ผู้ใช้สามารถแก้ไขเนื้อหาใน `foo.tex` แล้วสั่งคำสั่งดังนี้

```
% sgmltlatex -l foo.tex
```

คำสั่งนี้จะเรียก `cttex` มาตัดคำและส่งต่อให้ `latex` จัดการ. หลังจากนั้น `sgmltlatex` จะบันทึกไฟล์ `foo.tex` ในรูปที่ผู้ใช้สามารถแก้ไขได้อีก. ซึ่งหมายความว่าผู้ใช้ไม่มีความจำเป็นต้องสั่ง `cttex` และ `latex` พร้อมกันอีก.

เนื่องจากเอกสารแบบ LaTeX ที่ผลิตโดย SGML-tools ต้องการ `linuxdoc-sgml.sty`, `qwertz.sty`, `url.sty`, `epsfig.sty` และ `null.sty` ในการผลิตไฟล์ `.dvi` ผู้ใช้ต้อง configure โปรแกรม `latex` ให้รู้ว่าไฟล์เหล่านี้อยู่ที่ไหนด้วย. วิธีที่ง่ายที่สุดคือก็อปปี้ไฟล์เหล่านี้ซึ่งในกรณีของผู้เขียนจะอยู่ใต้ไดเรกทอรี่ /usr/local/lib/sgml-tools/ มาไว้ที่ไดเรกทอรี่ที่มีไฟล์ `.tex` อยู่หรือทำ symbolic link ก็ได้.

4 บทสรุป

การเขียนเอกสาร HOWTO ด้วยภาษา SGML และใช้ SGML-tools ในการแปลงเป็นเอกสารรูปแบบอื่นๆ ทำให้การเขียนเอกสารมีระบบระเบียบกว่าการเขียนเอกสารรูปแบบอื่นๆ. สะดวกในการแก้ไขเอกสารเพราะ

สามารถแก้จากไฟล์ SGML เดียวก็สามารถแก้เอกสารรูปแบบอื่นๆได้ด้วย. ถ้าหากมีเอกสารจำนวนมาก, ยังสามารถตีพิมพ์เป็นหนังสือได้ง่ายโดยใช้ LaTeX จัดการไฟล์ต่างๆ ซึ่งหนังสือเกี่ยวกับ HOWTO ของ Linux ก็ใช้วิธีนี้ในการพิมพ์เป็นหนังสือขาย.

การใช้ภาษาไทยพอจะทำได้ในระดับหนึ่งและต้องการการพัฒนาต่อไป.

5 แหล่งข้อมูลเพิ่มเติม

- *SGML-tools Homepage* `<http://www.sgmltools.org>`
- การใช้ภาษาไทยกับ *LaTeX* `<http://www.fedu.uec.ac.jp/ZzzThai/Linux/thailatex/>`
- `guide.sgml` และ `example.sgml` ซึ่งมากับ SGML-tools