

ConT_EXt + LuaT_EX + Chinese Howto

Kanru Chen < kanru@csie.us >

ConT_EXt 是比 L^AT_EX 年輕的 T_EX macro 套件，雖然說年輕，但是也已經有十幾年的開發歷史了。LuaT_EX 是從 pdfT_EX 和其他眾多計劃延伸出來的 T_EX 引擎，加入了 Lua(一個 script language)的支援，可以在 T_EX 裡面用 Lua 來寫程式，而且可以直接處理編譯過程中的各個環節。直接支援 Unicode 也是一個重點，另外一個很重要的功能是可以直接取用系統上的 OTF, TTF 字型。LuaT_EX 的主要開發者 Hans Hagen 也是 ConT_EXt 的主要開發者，所以這兩個東西就這樣很自然的結合起來了。我以前從來沒有用過 ConT_EXt，弄了半天終於讓 ConT_EXt、LuaT_EX、中文這樣東西可以一起跑起來了，這個 HowTo 就是結果，希望可以有多人來測試改進。:-)

安裝 ConT_EXt 跟 LuaT_EX

如果你跟我一樣是用 Debian 的話，在 /etc/apt/source.list 中加入以下路徑即可：

```
# for lmodern
deb http://people.debian.org/~preining/TeX/ lmodern/
deb-src http://people.debian.org/~preining/TeX/ lmodern/

# for context
deb http://people.debian.org/~preining/TeX/ context/
deb-src http://people.debian.org/~preining/TeX/ context/

# for luatex
deb http://people.debian.org/~preining/TeX/ luatex/
deb-src http://people.debian.org/~preining/TeX/ luatex/
```

安裝：

```
$ apt-get install context
$ apt-get install luatex
```

手動安裝的話，可以到 [ConT_EXt Download](#) 跟 [LuaT_EX Download](#) 去下載最新的版本之後，依照裡面的 INSTALL 檔案編譯安裝，其實 ConT_EXt 只需要把檔案解壓縮在 `~/texmf-local` 就可以了，跟安裝其他 T_EX macro 一樣。

設定 LuaT_EX

都安裝好後，設定以下環境變數，如：

```
$ export TEXMFCNF=/usr/share/texmf/web2c
$ export TEXMF='{/usr/share/texmf,/usr/share/texmf-texlive,/home/kanru/texmf}'
$ export TEXMFCACHE=/tmp
$ export OSFONTDIR='{/usr/share/fonts/truetype,/home/kanru/.fonts}'
```

其中 OSFONTDIR 是 LuaTeX 會去額外搜尋系統字型的路徑，目前不會遞迴搜尋，所以如果有多個目錄的話，請個別設定。

設定好後，執行：

```
$ texexec --make --lua en
```

這樣 ConTeXt 就會開始設定並預先產生需要的 cache。

測試

接下來可以先測試一下 LuaTeX 是不是真的已經可以用了，打開一個文件比如說 test.tex，裡面寫下：

```
\starttext  
Hello World!  
\stoptext
```

執行

```
$ texexec --lua test.tex
```

如果有產生 test.pdf 的話，那表示 LuaTeX 已經可以用了。

設定中文

接下來可以開始設定中文，首先要確定有字型可以用，以本文為例，是使用 cwTeX 的 cwheib.ttf 字型，如果這個字型有放在剛剛所設定的目錄中的話，用

```
$ luatools cwheib.ttf
```

應該可以找到這個字型的位置。

確定字型可以用之後，將以下設定貼入剛剛的 test.tex 中：

```

\enableregime[utf]
\usemodule[chi-00]
\directlua0{
  function wrap(str)
    rt = ''
    for u in string.utfvalues(str) do
      up = math.ceil(u / 0x100)
      low = u - math.ceil(u / 0x100) * 0x100
      if u < 127 then % need to be replace by cjk range
        rt = rt .. unicode.utf8.char(u)
      else
        rt = rt .. '\\lookaheaduchar{' .. up ..'}{' .. low ..'}'
      end
    end
    return rt
  end
  callback.register('process_input_buffer', wrap)
}
\beginLUA\insertunicodetwo
\unexpanded\def\insertunicodetwo
{\doinsertunicodetwo\unicodetwo}
\unexpanded\def\doinsertunicodetwo#1#2%
{\char\numexpr(#2+(#1*256))\relax}
\endLUA
\def\en#1{%
  \hspace\chinesesurroundskip
  \hspace\chineseinterglyphskip\relax
  #1%
  \hspace\chineseinterglyphskip\relax
  \hspace\chinesesurroundskip
  \hspace\chinesesurroundskip
  \hspace\chinesesurroundskip
  \hspace\chinesesurroundskip
  \hspace\chinesesurroundskip
  \hspace\chinesesurroundskip
  \hspace\chinesesurroundskip
  \hspace\chinesesurroundskip
  \ignorespaces
}
\starttypescript [sans] [cwheib] [name]
  \definefontsynonym [Sans] [cwheib]
  \definefontsynonym [SansItalic] [cwheib]
  \definefontsynonym [SansSlanted] [cwheib]
  \definefontsynonym [SansBold] [cwheib]
  \definefontsynonym [SansBoldSlanted] [cwheib]
  \definefontsynonym [SansBoldItalic] [cwheib]
\stoptypescript
\starttypescript [my] [cwheib]
  \definetypface [cwheib] [ss] [sans] [cwheib] [default]
\stoptypescript
\usetypescript [my] [cwheib]
\setupbodyfont [cwheib,ss,10pt]

```

接下來就可以輸入中文測試了。

未來工作

目前這個修改只是利用舊有處理中文的方法來處理中文的換行，中英字型間距等等，完全沒有動到 ConTeXt 的核心部份，事實上像是中英間距還有問題，是用不太巧妙的方法處理掉的，未來應該要用 Lua 重新架構中文處理的部份。

目前還有的問題有：

- 沒有辦法方便的切換中英字型(寫 macro 解決)。
- 中英間距自動調整錯誤。
- luatools 沒辦法遞迴搜尋字型(其實應該跟系統字型機制連結)。
- starttyping 也就是表列模式中沒辦法用中文(因為 macro 不會被展開)。
- LuaTeX 產生的 PDF 沒辦法複製貼上，不過這已在官方的修正計劃中。
- 沒有粗斜體的字型，就是沒有粗斜體。我忘了 L^AT_EX-CJK 是怎麼解決的了……好像是用模擬的？
- 把中文相關的修正整理為一個可以引用的 module 。

其他資源

想要開始寫 ConTeXt 或是開始 hack，可以參考以下資源：

1. 本文原始碼，裡面有些用法跟上面的範例不一樣。
2. [ConTeXt Wiki](#)，裡面有許多文件跟範例。
3. [LuaTeX 網站](#)，裡面有使用說明可以下載。
4. [Lua 網站](#)，要寫 Lua 可能需要參考一下手冊。

後記

研究 ConTeXt 是很有趣的事，看了這麼久，竟然也會看 TeX macro 了，macro 語言有趣的地方就是他會層層遞迴展開，而且可以重複更改某 macro 的定義，所以要修改某一功能，有時可以不用修改原程式碼，可以用重新定義 macro 的方式做到。

ConTeXt 某部份來說，其實是比較接近 Plain TeX 的，不需用到許多外部套件的特性，讓閱讀 ConTeXt 的程式簡單許多。

LuaTeX 的擴展性真的很不錯，看文件就可以知道有許多地方是可以 hook 的，未來利用這種特性，應該會有奇妙的發展。

2007年 八月 26日 version 0.1 陳侃如