Zpravodaj Československého sdružení uživatelů TeXu

Jiang Jiang

Chinese TeX Typesetting: Past and Present

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 20 (2010), No. 3, 215–219

Persistent URL: http://dml.cz/dmlcz/150123

Sazba čínštiny v TEXu: historie a současnost

Jjgod Jiang

Abstrakt

Článek seznamuje čtenáře s přehledem sazby čínštiny v TEXu od prvopočátků až do současnosti.

Klíčová slova: CCT, TianYuan, CJK, CTEX, XETEX, `xcp.py`, zhspacing, xeCJK, ctex, ctex-kit, ctex-doc, sazba čínštiny, Google code.

doi: 10.5300/2010-3/215

Introduction

Knuth developed TEX as a 7-bit system at the time it was invented, internationalization became a tough issue ever since then, yet CJK (Chinese, Japanese and Korean) typesetting is probably the toughest part. The biggest problem is that Eastern language systems like Chinese has a lot of characters compare to western languages like Latin, English and French.

There are more than 100 thousands of ideograph based characters and more than 10 thousand of which are used on daily basis. It means the typesetting system should be able to support such a large range of characters and use fonts accordingly. Unfortunately under a 7-bit or 8-bit system like Knuth-TEX, there is no trivial or elegance solution to support that.

CCT and TianYuan

Since the original TEX system does not provide such a capability, workarounds are invented. Early TEX adopters in China are mostly mathematicians, in 1990s, Zhang Linbo invented the CCT (`http://lsec.cc.ac.cn/~zlb/`) system while Xiao Gang and Chen Zhijie invented TianYuan system (TY for short; `http://wims.math.ecnu.edu.cn/ty/`).

Both of them are based preprocessors – using a preprocessing tool to convert from the original TEX source file in raw Chinese to a version that TEX can recognize, most importantly, the DVI generated in such methods can be further processed to use Chinese fonts. This mechanism was not very easy to use but worked fairly well. However, there are still many limitations:

1. preprocessing based solutions are cumbersome, you need an extra wrapper to the TEX executable otherwise files will be messed up;

2. font support is very limited, in both systems mentioned above, you can only use the fonts predefined, it is very difficult for them to support arbitrary fonts.

CJK package

Things changed a lot since Werner Lemberg introduced the CJK macro package (`http://cjk.ffii.org/`), its a pure TEX macro package so you don't need to preprocess the TEX source. Werner also defined a font system for CJK font selection, based on LATEX font selection scheme. To support fonts that contains thousands of glyphs, this system will need users to convert TTF files into multiple "sub-font" files (since TEX can support 256 font files at most, each of them can have 256 glyphs, so theoretically we can use 65536 different glyphs in one single TEX document), each one covers 256 different glyphs, combining this sub-font scheme and a PS/PDF generation driver that supports this scheme, we can correctly generate the resulting PDF (or PostScript). At that moment, DVIPDFM*x* (`http://project.ktug.or.kr/dvipdfmx/`) is such an PDF driver.

A typical CJK document looks like this:

```
\documentclass{article}
\usepackage{CJK}

\begin{document}
\begin{CJK}{GBK}{song}
你好，这里是一段中文内容。
\end{CJK}
\end{document}
```

As you can see, the text wrapped between `\begin{CJK}`... and `\end{CJK}` can be CJK characters, but everything outside cannot be. Obviously that is not very convenient since you will like to define macros containing CJK characters in the preamble from time to time.

This method works pretty well and proved to be quite stable, so more and more Chinese typesetters adopted it. Zhang Linbo also converted CCT to use a similar method, compare to CJK, CCT have more features focused on Chinese typesetting, such as white space adjustment, punctuation width adjustment, etc.

CTEX

As one memorable mark of TEX typesetting in China, Wang Lei and Wu Lingyun created a community called "CTEX" (`http://www.ctex.org/`), which later be-

came the most popular and most active community for TEX learning and development in China. CTEX was originally a mailing list, but converted to a Web forum later.

With the help of many others, Wu Lingyun packaged all mature Chinese TEX typesetting solutions with the most popular TEX distribution on Windows – MiKTEX – to an individual distribution called CTEX. CTEX became the most popular TEX distribution in China ever since then.

XƎTEX

However, with the above solutions, installing fonts for TEX usage is still very much a pain. It requires the users to convert fonts from TTF to TFM and edit mapping file very carefully. Thus, new TEX users consider font installation a very difficult part and try to avoid it as much as possible.

Something revolutionary happened since the introduction of XƎTEX created by Jonathan Kew (`http://scripts.sil.org/xetex`): processing Unicode documents became so simple and straightforward, especially for CJK documents: you can use any CJK characters in Unicode, not just limited to certain local character sets like GBK or Big5 or Shift-JIS. Best of all, you can use any system font directly, without doing any clumsy "subfont" conversion or carefully creating mapping files for PS/PDF drivers.

However, XƎTEX was only available to Mac users when it was first developed, because only a few TEX users use Mac at that time, it was not widely recognized. Soon after Jonathan Kew ported it to Linux and Akira Kakuto provided the Windows port (through `w32tex` and TEX Live), XƎTEX became very popular among Chinese users.

The low-level facility provided by XƎTEX is enough to typeset regular Chinese documents, but typesetters always want better results: switch fonts automatically between CJK and Latin text, adjust the width of punctuations to achieve better line breaking, better support for vertical layout are some of the common requests.

The `xcp.py` script and **zhspacing** package

Among these request, automatic font switching is especially urgent. At that time, most Latin glyphs from Chinese fonts are so badly designed that it is almost impossible to use. Naturally a typesetter would like to use the high quality Latin fonts for non-Chinese texts.

At first, Jiang Jiang designed a preprocessing script called `xcp.py` (`http://code.google.com/p/xcp/`), it simply wrap all the text in CJK with a macro to switch font. This solution is not very convenient but worked amazingly well for most Chinese documents, yet does not impose much performance penalty.

Since more and more users are requesting this feature, Jonathan Kew and Miyata Shigeru worked together to implement it as a build-in of XeTeX. Basically, you can assign a "character class" to each Unicode code point with `\XeTeXcharclass`, then define the command to invoke when switching between two specific class with `\XeTeXinterchartoks`.

With the sample code provided by Miyata Shigeru, Yin Dian implemented a package called zhspacing (`http://code.google.com/p/zhspacing/`), which has complete support to assign and switch fonts automatically for Chinese documents. zhspacing is a big step forward compare to `xcp.py`, and it is well documented in both English and Chinese.

xeCJK

Soon after the release of zhspacing, Sun Wenchang provided another package called xeCJK (`http://ctan.org/pkg/xecjk/`) with exactly the same purpose. xeCJK is more like a direct descendant of the CJK package by Werner Lemberg, users can port their CJK documents into xeCJK without much efforts, yet features like `CJKnumber` and `CJKpunct` remains compatible. xeCJK soon became the most popular solution for Chinese typesetting.

A typical document in xeCJK looks like this:

```
\documentclass{article}
\usepackage{xecjk}
\setmainfont{Palatino}
\setCJKmainfont{SimSun}

\begin{document}
你好，这里是一段中文内容。
\end{document}
```

While it is totally fine using xeCJK in your documents directly, it is still a bit inconvenient if you need to include other functions like captions ("Table of Contents", "Chapter", "Bibliography", and so on) in Chinese, punctuations in Chinese, hyperref with CJK bookmarks, etc. Different packages may have different options, which should be carefully tuned for each TeX engine and output drivers (pdfTeX / XeTeX, `dvipdfmx`, `dvips`).

ctex, ctex-kit and ctex-doc

To simplify this process, Wu Lingyun initiated a higher level package called ctex (`http://ctan.org/pkg/ctex/`), which contains several document classes and macro packages designed for Chinese typesetting with LaTeX.

ctex has multiple backends to support traditional CJK and pdfTEX based engine, or dvipdfmx based PDF output, while it also supports xeCJK and XHLATEX based engine. Fonts are also pre-defined in several schemes like "winfonts" and "adobefonts". A typical ctex document looks like this:

```
\documentclass{ctexart}

\begin{document}
你好，这里是一段中文内容。
\end{document}
```

The most recent effort of CTEX community is a project called ctex-kit (http://code.google.com/p/ctex-kit/), we organized all Chinese related TEX packages into a code repository on Google code, then we package and submit these packages to CTAN regularly. With this totally open source approach, these packages are now maintained in a more active state, and users can retrieve them easily from CTAN or through the package manager of TEX Live 2009.

Besides ctex-kit, there is another project called ctex-doc (http://code.google.com/p/ctex-doc/), which organized all of the translations (lshort-cn for instance) or original documents related to TEX, LATEX, ConTEXt and graphics tools like METAPOST.

Conclusion

With the collaboration by the whole Chinese TEX community, we have every reason to expect an even more bright future for Chinese TEX typesetting.

Summary: Chinese TEX Typesetting: Past and Present

The article introduces and overviews Chinese TEX typesetting from its early beginnings to the present day.

Keywords: CCT, TianYuan, CJK, CTEX, XHTEX, xcp.py, zhspacing, xeCJK, ctex, ctex-kit, ctex-doc, Chinese typesetting, Google code.

Jjgod Jiang, gzjjgod@gmail.com
6 South Kexueyuan Road
Beijing, 100080, China