

# 日本語の LuaTeX でのテスト

北川 弘典

2009/10/23

## 1 特徴

- 和文文字は全部 `\UMS{xxxxx}` の形に変換されて処理される（行末尾が和文文字ならば `\relax` を自動挿入する）．そのため、`\char` のように和文文字の文字コードを取得することはできないし、`\verb` コマンドなどで日本語を扱うことはできない．その一方、命令語に和文文字が来ても平気である：`LaTeX` とは `= LaTeX` とは
- LuaTeX のいつのバージョンで通るのかよくわからない．とりあえず僕は 0.43.0 を使っている．また、`\kern` による kern と `\ /` による kern を区別するため、添付した `luatex-0.43.0_italic.diff` を当てている．
- 和文文字の文字クラスの定義や禁則処理関連のパラメータは全て global で持っている．local に変更することはできない．
- 文字クラスの定義などは `ujisn.tfm` を参考にした．しかし、`\kanjiskip`、`\xkanjiskip` に対応するものは意欲がなぜかなかったので作ってない（調整するには `luaums.sty` を直接いじるしかない、ということ）し、和文フォントの読み込みや和文文字の `\UMS{xxxxx}` への変換時に使われるのでドキュメント中に変更することはできない．
- 禁則処理関係の `table` は global だが、文書中で変更することは差し支えないはずである（段落終了時の値が使われるはず）．変更用の `TeX` コマンドは用意してないが `ukinsoku-a.tex` (`upTeX` の `ukinsoku.tex` から生成) を見ればある程度はわかるかもしれない（Unicode でのコード番号を調べる必要があるが）
- 和文フォントの読み込みは `*.ttf` しか想定していない（`*.ttc` も OpenType Font も考えていない）．テストも `ipam.ttf`、`ipag.ttf` でしか行っていない．

以下は Wikipedia の  $\text{T}_{\text{E}}\text{X}$  の項目より (2009.10.23 閲覧)

## 2 $\text{T}_{\text{E}}\text{X}$

$\text{T}_{\text{E}}\text{X}$  (読み方は後述) は数学者・計算機科学者であるドナルド・クヌース (Donald E. Knuth) により作られた組版処理ソフトウェアである。

### 2.1 名称について

製作者であるクヌースによって以下のように要請されている。

#### 2.1.1 表記法

正しくは “ $\text{T}_{\text{E}}\text{X}$ ” と表記するが、それができない場合には “ $\text{TeX}$ ” と表記する (“ $\text{TEX}$ ” と表記するのは誤り)。

#### 2.1.2 読み方

$\text{T}_{\text{E}}\text{X}$  はギリシャ文字の  $\tau$  (タフ・エプシロン・ヒ) であるから、「テックス」ではなく、ギリシャ語読みの [tex] (「テフ」) のように発音するのが正しい。しかしそのような発音は難しいので、クヌースは「テック」と読んでも構わないとしている。日本では「テフ」または「テック」という読み方が広まっている。

### 2.2 機能

$\text{T}_{\text{E}}\text{X}$  はマークアップ言語処理系であり、文章そのものと、文章の構造を指定する命令とが混在して記述されたテキストファイルを読み込み、そこに書かれた命令に従って文章を組版して、組版結果を DVI 形式のファイルに書き出す。DVI 形式というのは、装置に依存しない (device-independent) 中間形式である。

DVI ファイルには紙面のどの位置にどの文字を配置するかといった情報が書き込まれている。実際に紙に印刷したりディスプレイ上に表示したりするためには、DVI ファイルを解釈する別のソフトウェアが用いられる。DVI ファイルを扱うソフトウェアとして、各種のビューワや PostScript など他のページ記述言語へのトランスレータ、プリンタドライバなどが利用されている。

組版処理については、行分割およびページ分割位置の判別、ハイフネーション、リガチャ、およびカーニングなどを自動で処理でき、その自動処理の内容も種々のパラメータを変更することによりカスタマイズできる。数式組版についても、多くの機能が盛り込まれている。 $\text{T}_{\text{E}}\text{X}$  が文字などを配置する精度は  $[25.4/(72.27 \times 65,536)]$  mm (約 5.363 nm、4,736,286.72 dpi) である。

$\text{T}_{\text{E}}\text{X}$  の扱う命令文の中には、組版に直接係わる命令文の他に、新しい命令文を定義するための命令文もある。 $\text{T}_{\text{E}}\text{X}$  のこの機能を使って使用者が独自に作った命令文はマクロと呼ばれ、こうした独自の改良をマクロパッケージと呼ばれる形で配布できる。

比較的よく知られている  $\text{T}_\text{E}\text{X}$  上のマクロパッケージには、クヌース自身による plain  $\text{TeX}$ 、一般的な文書記述に優れた  $\text{L}^{\text{A}}\text{T}_\text{E}\text{X}$ 、数学的文書用の  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{T}_\text{E}\text{X}$  などがある。一般の利用者は、 $\text{T}_\text{E}\text{X}$  を直接使うよりも、 $\text{T}_\text{E}\text{X}$  に何らかのマクロパッケージを読み込ませたものを使うことが多い。そのため、これらのマクロパッケージのことも “ $\text{T}_\text{E}\text{X}$ ” と呼ぶ場合があるが、本来は誤用である。

$\text{T}_\text{E}\text{X}$  のマクロパッケージには、他にも参考文献リストの作成に用いる  $\text{BibT}_\text{E}\text{X}$ 、プレゼンテーション用スライドの作成に用いる  $\text{SLiT}_\text{E}\text{X}$ 、数学的文書の記述に強い  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{T}_\text{E}\text{X}$  の機能と  $\text{L}^{\text{A}}\text{T}_\text{E}\text{X}$  の機能を併せ持った  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-L}^{\text{A}}\text{T}_\text{E}\text{X}$ 、化学構造式の描画に用いる  $\text{X}^{\text{M}}\text{T}_\text{E}\text{X}$ 、楽譜の記述に用いる  $\text{MusixT}_\text{E}\text{X}$  などがある。

$\text{T}_\text{E}\text{X}$  とそれに関連するプログラム、および  $\text{T}_\text{E}\text{X}$  のマクロパッケージなどは CTAN (Comprehensive  $\text{T}_\text{E}\text{X}$  Archive Network、包括  $\text{T}_\text{E}\text{X}$  アーカイブネットワーク) からダウンロードできる。

## 2.3 生い立ち

$\text{T}_\text{E}\text{X}$  は、クヌースが自身の著書 *The Art of Computer Programming* を書いたときに、組版の汚さに憤慨し、自分自身で心行くまで組版を制御するために作成したとされている。開発にあたって、伝統的な組版およびその関連技術に対する広範囲にわたる調査を行った。その調査結果を取り入れることで、 $\text{T}_\text{E}\text{X}$  は商業品質の組版ができる柔軟で強力な組版システムになった。

$\text{T}_\text{E}\text{X}$  はフリーソフトウェアであり、ソースコードも公開されていて、誰でも改良を加えることができる。その改良版の配布も、 $\text{T}_\text{E}\text{X}$  と区別できるように別名を付けさえすれば許される。また、 $\text{T}_\text{E}\text{X}$  は非常にバグが少ないソフトウェアとしても有名で、ジョーク好きのクヌースが、バグ発見者に対しては前回のバグ発見者の 2 倍の懸賞金をかけるほどである。この賞金は小切手で払われるのだが、貰った人は記念に取っておく人ばかりなので、結局クヌースの出費はほとんど無いという。

クヌースは  $\text{T}_\text{E}\text{X}$  のバージョン 3 を開発した際に、これ以上の機能拡張はしないことを宣言した。その後は不具合の修正のみがなされ、バージョン番号は 3.14、3.141、3.1415、... というように付けられている。これは更新のたびに数字が円周率に近づいていくようになっていて、クヌースの死の時点をもってバージョン  $\pi$  として、バージョンアップを打ち切るとのことである。

クヌースは  $\text{T}_\text{E}\text{X}$  の開発と同時に、 $\text{T}_\text{E}\text{X}$  で利用するフォントを作成するためのシステムである METAFONT も開発した。こちらのバージョン番号は 2.71、2.718、2.7182、... というように、更新のたびに数字がネイピア数に近づいていくようになっている。さらに、クヌースは METAFONT を使って、 $\text{T}_\text{E}\text{X}$  上で使われるフォントの Computer Modern のデザインも行った。

$\text{T}_\text{E}\text{X}$  および METAFONT は、これもクヌース自身によって提唱されている文芸的プログラミング (Literate Programming) を実現する WEB というシステムで Pascal ヘトランスレートされることを前提に記述されている。しかし実際には WEB2C で C 言語に変換してコンパイルされ実行形式を得ることが多い。

## 2.4 $\text{\TeX}$ の日本語化

日本語組版処理のできる日本語版の  $\text{\TeX}$  および  $\text{\LaTeX}$  には、アスキー・メディアワークスによる  $\text{p}\text{\TeX}$  および  $\text{p}\text{\LaTeX}$  と、NTT の斉藤康己による  $\text{NTT}\text{\jTeX}$  および  $\text{NTT}\text{\jLaTeX}$  などがある。

$\text{\TeX}$  の日本語対応において技術的に最も大きな課題は、複数バイト文字コードへの対応である。 $\text{p}\text{\TeX}$  (および前身の日本語  $\text{\TeX}$ ) は、JIS X 0208 を文字集合とした文字コード (ISO-2022-JP、EUC-JP、および Shift\_JIS) を直接扱う。DVI フォーマットは元々 16 ビット以上の文字コードを格納できる仕様が含まれていた。しかしオリジナルの英語版では使われていなかったため、既存プログラムの多くは  $\text{p}\text{\TeX}$  が出力する DVI ファイルを処理できない。またフォントに関係するファイルフォーマットが拡張されている。これに対して  $\text{NTT}\text{\jTeX}$  は、複数の 1 バイト文字セットに分割することで対応している。例えば、ひらがなとカタカナは内部的には別々の 1 バイト文字セットとして扱われる。このためにオリジナルの英語版からの変更が小さく、移植も比較的容易である。ファイルフォーマットが同じなので英語版のプログラムで DVI ファイル等を処理することもできる。しかし後述するフォントのマッピングの問題があるため、実際には多くの使用者が  $\text{NTT}\text{\jTeX}$  用に拡張されたプログラムを使っている。

使用する日本語用フォントについては  $\text{p}\text{\TeX}$  が写研フォントの使用を、 $\text{NTT}\text{\jTeX}$  が大日本印刷フォントの使用を前提としており、それぞれフォントメトリック情報 (フォントの文字寸法の情報) をバンドルして配布している。しかし有償であるこれらのフォントのグリフ情報を持っていなくても、画面表示や印刷の際に使用者が利用できる他の日本語用フォントで代用することができる。つまり写研フォントや大日本印刷フォントのフォントメトリック情報を用いて文字の位置を固定し、画面表示や印刷には他の日本語用フォントを用いていることが可能である。このため日本語化された  $\text{\TeX}$  関係プログラムのほとんどは、画面表示や印刷で実際に使うフォントを選択できるように、フォントのマッピング (対応付け) を定義する機能を持っている。

歴史的には、アスキーが日本語  $\text{\TeX}$  の PC-9800 シリーズ対応版を販売したために個人の使用者を中心に普及した。一方、 $\text{NTT}\text{\jTeX}$  は元の英語版プログラムからの変更が比較的小さいという利点を受けて、UNIX および UNIX 互換 OS を使う大学や研究機関の関係者を中心に普及した。

しかし現在では次に挙げる理由から、日本語対応  $\text{\TeX}$  として  $\text{p}\text{\TeX}$  が使われていることが多い。

- UNIX 用、および UNIX 互換 OS 用の主な日本語対応  $\text{\TeX}$  配布形態である  $\text{ptexlive}$  や  $\text{ptetex3}$  が  $\text{p}\text{\TeX}$  のみを採用している。
- Microsoft Windows 用の主な日本語対応  $\text{\TeX}$  配布形態である  $\text{W32}\text{\TeX}$  が  $\text{p}\text{\TeX}$  を扱える ( $\text{NTT}\text{\jTeX}$  も扱える)。
- $\text{p}\text{\TeX}$  の扱い方を解説する文献の方が、 $\text{NTT}\text{\jTeX}$  のものに比べて、出版物と Web 上文書の両方で多い。
- $\text{p}\text{\TeX}$  は縦組みにも対応しているが、 $\text{NTT}\text{\jTeX}$  は対応していない。

(以下略)