# CJK パッケージによる LaTeX の日本語化と LaTeX2HTML との関係

平成 15 年 2 月 10 日

情報電子工学科 竹野研究室
山岡 薫

# 目 次


**1 はじめに** **1**

**2 いろいろな TEX** **1**

2.1 TEX . . . . . 1

2.1.1 TEX と日本語 . . . . . 2

2.1.2 TEX とマクロパッケージ . . . . . 3

2.2 LATEX . . . . . 4

2.3 LATEX 2ε . . . . . 6

2.4 LATEX(LATEX 2ε) の使用例 . . . . . 6

**3 日本語 TEX, NTT 版 jTEX, CJK パッケージ** **8**

3.1 日本語表現の方法 . . . . . 8

3.2 アスキー版 pTEX について . . . . . 8

3.3 NTT 版 jTEX について . . . . . 9

3.4 CJK パッケージについて . . . . . 10

3.5 アスキー日本語 pTEX と NTT 版 jTEX . . . . . 11

**4 LATEX2HTML** **13**

4.1 HTML と LATEX の関係 . . . . . 13

4.2 LATEX2HTML の概要と特徴、必要なもの . . . . . 13

4.3 各種言語への対応 . . . . . 15

4.4 LATEX2HTML の日本語化 . . . . . 17

4.5 LATEX と HTML との相互変換について . . . . . 18

**5 対応と比較、検討** **18**

5.1 CJK パッケージと pTEX の比較 . . . . . 18

5.2 欧文アクセントと出力漢字コードの問題 . . . . . 20

5.3 アクセント問題の実験 . . . . . 21

5.4 実験結果についての考察 . . . . . 23

**6 まとめ** **24**

**参考文献** **25**

## 概要

本稿は、 CJK パッケージによる LaTeX の日本語化と LaTeX2HTML の日本語化の関係について考察する。まず、 TeX と LaTeX2HTML に関する特徴などを述べ、 LaTeX で日本語を表現するアスキー版 pTeX, NTT 版 jTeX, CJK パッケージの差異、及び LaTeX2HTML への対応の現状と比較、検討を行い、そして LaTeX2HTML とその日本語化に関する問題点を、日本語漢字コードと欧文アクセントに関する問題を中心に考察し、LaTeX2HTML の日本語化では、それらをどう処理すべきかなどを提案する。

# 1　はじめに

現在 OS の代表と質問されると Windows や Macintosh と言った返答が大半で、 UNIX という返答は少ないだろう。研究室に配属されるまでは OS の中に UNIX に含まれていることや、使い方すら分からなかった。しかし使い慣れてくると Windows より UNIX の方が機能面が使いやすいと感じた。慣れ始めた時に、担当教員の情報電子工学科 竹野茂治 助教授の話から LaTeX2HTML を使用して CJK パッケージによる LaTeX の日本語化についてを研究として取り上げた。

# 2　いろいろな TeX

## 2.1　TeX

TeX はスタンフォード大学の Donald E.Knuth 元教授が作成したコンピュータ組版システムで、言い換えるとコンピュータの種類やハードに依存せずに原稿の組み版を作成するソフトである。これはもともと欧文用のソフトウェアなので日本語は使用できないが、アスキー社が欧文用 TeX に手を加え日本語を使えるようにしたアスキー日本語 TeX(pTeX) などがある。
ここで TeX の特徴と組版規則について述べる。まず特徴には、

- 美しい仕上がり
- 高度な数式組版能力
- マクロ機能による拡張性
- 多様な出力環境への対応
- マークアップ方式での組版指定
- オープンソース
- 出版のための組版費用や期間の削減

などが挙げられる。オープンソースとは、ソフトウェア開発手法のひとつでソフトウェアのソースプログラムを公開することにより、利用する多数の人によってソフトウェアの欠陥を発見もしくは修正し、足りない機能や改良点を補足できる。出版のための組版費用や期間の削減とは、 TeX の場合は、入力はエディタ (文書入力専門) が用意されていて保存用に DVI ファイルがあり、出力には専用のドライバーがある。TeX はフリーソフトなのでネットワークで自由に配布されているのと、マクロが安価で入手できること、処理が高速なので編集機能も優れていることや時間が短縮できることに対してワープロの場合はキーボードから入力すると同時に印刷結果のイメージを画面に表示する方式なので反応が TeX に比べて遅い。しかしフリーソフトは無料で入手できるかわりに、ソフトの使用に関してはファイルの破損など全て自分で責任を取らないといけないし、メーカーのサポー

トがないので処理は全て自分で行わなければいけない等のデメリットもある。
次に TEX には以下のような組版規則が組み込まれている。

1. プロポーショナル・スペーシング
2. 合字 (ligature)
3. カーニング
4. 最適な行割りとページ割り
5. ハイフネーション
6. 数式の組版

プロポーショナル・スペーシングとは 1 文字当りの横幅をバランスのとれた幅にすることである。合字とはクラシックな感じをだすために ffi や Æsop's などと表現することを意味している。カーニングとは簡単に字詰めのことを意味している。最適な行割りとは段落全体にわたって字の詰まり具合をなるべく均一にしてくれることで、最適なページ割りとは見出しの直後でページが割れないようにしてくれるということである。ハイフネーションとは、どうしても行割りがうまくできないときに、単語を音節の区切りで割り、行末にハイフン (-) をつけてくれることである。

### 2.1.1 TEX と日本語

TEX は英語での処理を対象にしているので日本語の使用ができない。扱うようにするには、

- 2 バイトのコード系の扱い
- 和文フォントの処理

を考えるが、お互い密接な関係があるので別々に扱うことはできない。TEX は 1 バイトコード用に設計されているので 2 バイトのコードは扱えず、フォントにおいても 1 フォントに 256 文字までしか収めることができないが、日本語は 2 バイトコードであり文字の種類も 256 文字ではおさめることができない。解決する方法として、

1. 和文フォントを 256 文字以下に分割して扱う
2. プログラム、フォントファイル等を変更

がある。 1. の場合は、何種類もある漢字の文字を 256 文字以下に分割し、フォントと 1 バイトのコードで実際の文字を表現する方法で、特徴は

- 変更が比較的少ない

- フォントの種類に制限がある
- フォントの整合性がとれない

が挙げられる。フォントの整合性がとれないというのは、本来 1 つあるべきフォントが複数に分かれてしまうので、カーニング処理など同一フォント内でのみサポートされている機能を使えなくしてしまう。 2. の場合も、プログラムやフォントファイル自体を日本語に合わせて変更していく方法で、特徴は、

- 作業量が膨大である
- フォント使用の制限がない
- 機能を追加できる

が挙げられる。フォント使用の制限がないとは、和文フォントは 1 フォントにすべての文字をふくんでいるので、欧文フォントと合わせて 256 フォントまで対応していて、一度に使えるフォントの種類には制限がないという意味である。TeX で使える和文フォントがなかったので、何種類もの和文フォントの使用のために日本語に合わせていく方法を選んで、TeX が 2 バイト系のコードに対応するようにプログラムに手を加えて、 1 バイトコードの文字と 2 バイトコードの文字を、 TeX から同じ扱いできるようにしたものが日本語 TeX である。この日本語 TeX にはアスキー日本語 TeX と NTT 版 jTeX 等があるが、 1. NTT 版jTeX で 2. はアスキー日本語 TeX である。これらについては後で述べる。 日本語の文章作成にも以下のような処理特徴がある。

- 括弧類や句読点が連なった場合は、文字の組合せによって文字幅を変更できる
- 行頭に句読点や感嘆符、疑問符等を付けない
- 行末に括弧類を付けない
- 段落の最後の行を「す。」のように一字だけ残さないようにする

### 2.1.2 TeX とマクロパッケージ

最も狭い意味での TeX、つまり基本的な組版機能のみを持っている TeX には、文字の種類の変更等の基本的な機能はあるが、章の見出しを作るなどの機能はない。しかし、この TeX に色々な種類のマクロパッケージを組み合わせることによってそのような機能を使用できるようになる。ここでマクロとは、 TeX に追加登録する機能があって、この追加登録した機能のことを指す。

例えば、TeX に plainTeX マクロパッケージを組み込むと plainTeX となる。

TeX+ plainTeX マクロパッケージ = plainTeX

また、この plainTEX に LATEX マクロパッケージを組み込むと、

plainTEX+ LATEX マクロパッケージ = LATEX

となる。

## 2.2 LATEX

LATEX は、 DEC 社 (現在は Hewlett-Packard 社) の Leslie Lamport 氏によって開発された TEX のマクロパッケージで、解りやすく言えば TEX を基本に改良した文書処理システムである。 LATEX の特徴には、

- セクションやサブセクション等のマクロ構造と視覚的なレイアウトとを分けるようなマクロが用意
- 章・節・図・表・数式などの番号を自動的に付けてくれる
- 参照箇所には番号やページの挿入を自動的に行う
- 目次や索引、引用文献の処理を自動的に行う

などが挙げられる。

次に LATEX に伝えるコマンドを下記に挙げてみる。

1. スペース
   LATEX では、スペースやタブといったホワイトスペースは、どちらも同じようにスペースとして扱い、ホワイトスペースとして扱う場合は、 2 つ以上が連続していても 1 つのスペースとして扱われる。入力ファイルにおける行頭のホワイトスペースは普段使用するときは無視され、行末の改行はスペースと同様に処理される。

2. 特殊文字
   LATEX では、以下の記号は特別な意味を持っていて、フォントにおいて直接打ち出すことができない特殊文字になっている。

   ```
   $   &   %   #   {   }   ~
   ```

   これらを打ち出すためには、

   ```
   \$  \&  \%  \#  \{  \}  \~
   ```

   としなくてはいけない。

3. LATEX コマンド
   LATEX コマンドは、大文字小文字の区別があり、以下の 2 つの形式を採る。

- コマンドはバックスラッシュで始まり、アルファベットのみの文字列である。各コマンド名は、スペース、数字もしくはアルファベット以外の文字によって区切られている。
- コマンドはバックスラッシュと 1 つの特殊文字から作成する

LaTeX は、コマンドのあとのホワイトスペースは無視するので、スペースを入力するときはコマンド名のあとに {} とスペース、もしくはスペースコマンドを置かなければいけない。

4. コメント
   入力ファイルの中に % があると、 LaTeX はそこから行末までの文字、改行文字そして次の行頭のすべてのホワイトスペースを無視する。また、 % はホワイトスペースや改行が許されない長い入力行を分割するためにも使用できる。

5. 入力ファイルの構造
   LaTeX の入力ファイルを処理する際、最初に LaTeX に動作を指示する命令として、

```
\documentclass{...}
```

   で始め、文書全体のスタイルを変更するコマンドやシステムに新しい機能を追加するパッケージを読み込んだりするために

```
\usepackage{...}
```

   コマンドを使用する。このあとは文書全体を

```
\begin{document}
   文書
\end{document}
```

   で囲む。

**plainTeX の持つ機能** plainTeX コマンドは LaTeX で使用可能だが、 plainTeX 内でコマンドの問題を引き起こすかどうかは、コマンド群が集まって 1 つのシステムとして機能するように LaTeX では行われているので注意する。LaTeX でも有効な plainTeX の特徴には

- 数式を美しく作成
- 文書の最初はバックスラッシュで始まる命令
- 左右が揃うように文を詰めてくれる
- 段落は空行で区切る
- 段落の最初を字下げしてくれる

- ページ番号を付けてくれる

などがある。

## 2.3 LaTeX 2ε

LaTeX 2ε は Frank Mittelbach 率いる LaTeX3 のプロジェクトチームが様々な改良と今までリリースされて以来生じていたすべての修正を再統合して LaTeX を改良したソフトである。ここで追加された新しい機能を以下に述べる。

1. SLITEX の統合
2. アウトプット・ルーチンの改良
3. スタイルファイルに対するインターフェースの整理
4. box コマンドの拡張
5. \newcommand の拡張
6. bezier.sty の機能の取込み
7. カラー、グラフィックス関連の標準パッケージの提供

LaTeX 2ε は LaTeX を命令の記述方式など大幅に改良したもので、特殊なものでない限り従来のデータを利用できるように LaTeX にも対応している。

## 2.4 LaTeX(LaTeX 2ε) の使用例

TEX(LaTeX,LaTeX 2ε,pLaTeX) を使用すると種々な書体や文字サイズの変更などで美しく印刷できる。
例えば、

$$y = ax^2 + bx + c$$

$$(a+b)^2 = a^2 + 2ab + b^2$$

$$(a+b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$$

など簡単な数式や、

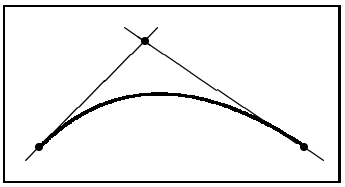
$$\left( \int_0^\infty \frac{\sin x}{\sqrt{x}} dx \right)^2 = \sum_{k=0}^{\infty} \frac{1}{2k+1} \frac{(2k)!}{2^2 k \left(k!\right)^2} = \prod_{k=1}^{\infty} \frac{4k^2}{4k^2-1} = \frac{\pi}{2}$$

$$V = \begin{pmatrix} \frac{a^0}{\sqrt{2}} + \frac{b^2}{\sqrt{6}} & a^+ & x_{13} \\ a_{12} & \frac{-a^0}{\sqrt{2}} + \frac{b^0}{\sqrt{6}} & E^0 \\ E^- & x_{32} & \frac{d_1}{c_1 + \frac{d_2}{c_2}} \end{pmatrix}$$

のような複雑な数式でも簡単に出力できるし、

“¿But aren’t Kafka’s Schloß and Æsop’s Œuvres often naïve vis-á-vis the dæmonic phœnix’s official rôle in fluffy soufflés?”

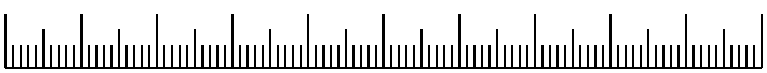
のような複雑な欧文を打つことや滑らかな 2 次ベジェ曲線 (quadratic Bézier curve) を描画することも描いてみると、



となり、このソースは下記に示す通りである。

```
\begin{center}
\fbox{\begin{picture}(120,60)
  \put(10,10){\circle*{3}}
  \put(50,50){\circle*{3}}
  \put(110,10){\circle*{3}}
  \thinlines
  \put(5,5){\line(1,1){50}}
  \put(117.5,5){\line(-3,2){75}}
  \thicklines
  \qbezier(10,10)(50,50)(110,10)
\end{picture}}
\begin{center}
```

次に定規図を描いてみると、



となり、このソースは、

```
\begin{center}
 {\unitlength=1cm
 \begin{picture}(10,0.7)
   \put(0,0){\line(1,0){10}}
   \multiput(0,0)(1,0){11}{\line(0,1){0,7}}
   \multiput(0,0)(0.5,0){20}{\line(0,1){0.5}}
   \multiput(0,0)(0.1,0){100}{\line(0,1){0.3}}
 \end{picture}}\\
```

である。

## 3 日本語 TEX, NTT 版 jTEX, CJK パッケージ

### 3.1 日本語表現の方法

TEX(LATEX, LATEX 2ε) において日本語表現する方法は、

1. ASCII (アスキー) 版 pTEX
2. NTT 版 jTEX
3. CJK パッケージ

の 3 通りある。
ここで アスキー 版 pTEX は、アスキー社が開発した日本語 TEX を更に改良し縦組ができるようにしたものである。そして NTT 版 jTEX は、 NTT の齋藤康己さんが開発し、千葉大学の桜井貴文さんが UNIX として使用できるようにされたものである。CJK パッケージは、 LATEX 上で英語だけでなくパッケージとして中国語、日本語、そして韓国語を使用できるするためのマクロである。

### 3.2 アスキー版 pTEX について

pTEX は株式会社アスキーが TEX を日本語化し、 p は publishing(出版) という意味である。これは縦組できる機能がある。追加された日本語の機能は、

- 横組と縦組の混在 ··· 横組だけでなく縦組でも文章を組版できる
- 日本語禁則処理 ··· 句読点や括弧 (かっこ) 類の禁則処理に対応している
- 文字間スペース ··· 和文と和文、和文と欧文の間のスペースを調整することで、高い品質を得られる
- JFM による和文文字組版 ··· 日本語の文字組版に JFM ファイルを用いる

などである。他にも pTEX には、

- \tate,\yoko ··· 組版方向を変更する命令
- \iftdir,\ifydir ··· 現在の組版方向を調べる命令
- \iftbox,\ifybox ··· 箱の組版方向を調べる命令
- \ybaselineshift,\tbaselineshift ··· 横組と縦組の欧文ベースラインの移動量
- Q,H ··· 長さの単位

のようなマクロでない命令 (プリミティブ) が追加されている。

### 3.3 NTT 版 jTeX について

NTT 版 jTeX は日本語を扱える TeX であり、オリジナルの TeX の枠組を超えるような大幅な改造は加えずに、互換性を意識した日本語化を行っているので、海外で作られた多くのソフトウェアも簡単に使用できる。日本語処理についてはアスキー日本語 TeX と若干異なり DVI ファイルに互換性がない。これは両者の DVI ファイルに互換性がないという意味で、オリジナルの TeX との関係を述べたものではなく、 NTT 版 jTeX の DVI ファイルはオリジナルの TeX との互換性はあってもアスキー日本語 TeX のほうは互換性がないという意味である。しかし LaTeX 上でのマークアップだけ行うなら両者の違いは感じることはないが、 NTT 版では和文用スタイルファイルは j-article.sty のように j の後にハイフンが付き、文章ファイルの最初の行には、

```
\documentclass [12pt] {j-article}
```

となる。文書スタイルのオプション a4j, b4j, b5j もそれぞれ a4, b4, b5 となる。NTT 版 jTeX の欠点は、日本語のフォントがたくさん使えないことである。

日本語の入出力 TeX ファイルでは使用する日本語の文字コードを指定して置かなければならない。よく使われる漢字コードについて述べると NTT 版 jTeX の場合は、 JIS, シフト JIS、EUC の漢字コードに対応している。その漢字コードの設定は以下で行うことができる。

- 端末表示用漢字コード ··· \kanjiterminaltype
- ファイル出力用漢字コード ··· \kanjifiletype

これはインストール時に初期設定されているが、設定変更は可能である。ここで以下に変更可能な部分を述べる。

- 入力 : 行を読み込む時に \kanjifiletype に指定されているコードを受け付けるが、端末から入力する場合は \kanjiterminaltype の指定が行われる。どのコードにも JIS コードが必ず受けつけられ、また JIS コードが設定されている場合は EUC も受けつける。

- 端末への出力 : アスキー文字から日本語に変更されるときに \kanjiterminaltype に設定されているコードが使用される。
- log ファイルへの出力 : アスキー文字から日本語に変更されるときに \kanjifiletype に設定されているコードが使用される。
- \write による出力 : その出力ファイルを \openout したときに \kanjifiletype に設定されているコードが使用される。

漢字コードについては NTT 版 jTEX の場合は、 JIS, シフト JIS、EUC の漢字コードに対応している。

日本語の文字間隔 日本語の文字間隔は \ja と呼ばれるサブフォントの \fontdimen というパラメタ群に設定されている。 \fontdimen は各フォントに関連した情報が収めてあるファイルで、英語の場合は単語間の空白の長さ等のパラメタがある。

禁則文字処理 この設定は jtexplains/jtexbase.tex で行われている。文字の直前で改行しなければいけない時に \prekinsoku を使用し、文字の直後での改行の場合は \postkinsoku を使用する。単純な縦書きマクロでは nokinsoku.sty の \neverkinsoku というコマンドを使用する。

日本語フォント NTT 版 jTEX での日本語フォントは明朝体 (dm) とゴシック体 (dg) が標準で使うことができる。これは \jfont というマクロを使って指定できる。

### 3.4 CJK パッケージについて

CJK パッケージの CJK とは、

C = China

J = Japan

K = Korea

という意味を表している。これはパッケージとして中国語と日本語、韓国語を使用して、漢字圏の異なる言語をファイル内の定義を変更することで同一画面に同時に表示させるためのものである。
CJK パッケージを使用しての LATEX の場合は、

```
\documentclass[a4paper]{article}
\usepackage[global]{CJK}
\{indentfirst}
\{document}
\{CJK*}[dnp]{JIS}{goth}

日本語の文章
```

```
\end{CJK*}
\end{document}
```

となる。
CJK パッケージは CJK と CJK* という 2 つの環境を与える。CJK* 環境は CJK 文字の後ろの空白を圧縮し、逆に CJK 環境のほうは圧縮しない違いがある。日本語のみの使用であれば CJK* 環境のほうを使用し、この中で CJK の文字の後に空白が必要な時は '\' や '{ }' などの TeX コマンドを使用する。また、日本語以外の言語の中に日本語を書けるようにするには、

```
\usepackage[global]{CJK}
```

と付け加える。

### 3.5 アスキー日本語 pTeX と NTT 版 jTeX

**概要** NTT 版 jTeX とアスキー日本語 TeX は環境や設計方針が異なっている。 NTT 版 jTeX は、研究者自身の研究環境の整備が目的で最小限で日本語が使用できるために作成されたものに対して、アスキー日本語 TeX は、日本語の書籍や雑誌の出版を行う目的に TeX をベースに日本語の出版システムの開発として開発されたものである。

**特徴** NTT 版 jTeX は dviware (プレビューやプリンタドライバなど) を改良しなくても、英語版のままで使用できることが多いこと、逆にアスキー版日本語 TeX は縦書きができたり多様なフォントが扱えるという特徴を持っている。

**jTeX と pTeX の差異** 分かりやすい一番大きな違いはファイルを作成しコンパイルして表示させるときの DVI ファイルの構造である。他にどのようなところの違いがあるのか詳細に述べる。

1. 文字サイズ、行送り pTeX と jTeX の文字サイズ、行送りについて述べる。
   - pTeX の場合
     ペナルティー (減点) によって行われている。例えば閉じ括弧 ')' が文の行頭に現われるのを防ぐために、閉じ括弧の直前で改行すると 800 点減点というようになっている。そして、文章での段落ごとにペナルティの合計が最小の点数になるように行分割されるようになっている。これは kinsoku.tex というファイルの中で、
     (a) \prebreakpenalty ')'=800 ··· 直前の改行に対する減点
     (b) \prebreakpenalty '('=800 ··· 直後の改行に対する減点
     と初期設定されているが、得点の変更は可能である。

- jTEX の場合
  欧文文字と和文記号、平仮名、片仮名については、行頭行末の禁則処理を行うことができる。例えば小さい『っ』を行頭に持ってこないようにするには作成中に、
  \prekinsoku‘っ
  と指定する。 pTEX とは異なり jTEX は、禁則の度合いを指定することはできないが、直前の和文文字との間での文字間隔を挿入しないことになっている。

2. 文字間隔の挿入 pTEX と jTEX の文字間隔の挿入について述べる。
   - pTEX の場合
     グルーの挿入を禁止・許可する以下の命令がある。
     (a) \noautospacing · · · 全角文字間のグルー挿入を禁止
     (b) \autospacing · · · 全角文字間のグルー挿入を許可
     (c) \noautoxspacing · · · 全角半角間のグルー挿入を禁止
     (d) \autoxspacing · · · 全角半角間のグルー挿入を許可
   - jTEX の場合
     和文文字間には \jintercharskip というグルー (伸縮する空白) が自動的に入る。これはファイルの中で、

     ```
     \jintercharskip= 0pt plus 0.25pt minus 0pt
     ```

     で指定されている。しかしポイント数字の変更は可能。このグルーは行頭禁則文字の直前、または行末禁則文字の直後には入らない。欧文文字と和文文字の間には \jasciikanjiskip と言うグルーが挿入され、数式と和文文字の間には \jathkanjiskip というグルーが挿入される。この 2 つの間隔は全角文字幅の 1/4 にしてある。

3. 数式中の和文
   従来の NTT 版 jTEX では数式中に和文を使うには \mbox · · · で囲まなければいけなかったが、現在は囲む必要はない。しかし現在も添字に和文文字を使っても小さくならない。

4. 改行の扱い
   文書を書く上で改行の扱いを特に注意しなければいけない。 pTEX も jTEX も和文文字の直後の改行を無視するが、jTEX では何も指定しなければ行頭が和文文字の時は、直前の改行も無視される。しかし、改行の直前が欧文のコンマかピリオドの時のみ、その改行は無視されない。よって改行を無視するのかどうかは 1 文字先読みしなければならない。設定は jTEX の場合は変更できる。

5. システム上の違い

pTeX の場合は jarticle とフォント指定することで欧文、和文とも出力できるのに対して、 jTeX の場合は j-article のように j の後にハイフンを付けて指定する。 NTT 版 jTeX の日本語の入出力において、漢字コードの設定は表示用とファイル用に設定し、その値は下記のようになっている。

| jTeX | 漢字コード |
|---|---|
| 0 | ASCII 文字のみ |
| 1 | JIS(ESC-$-@, ESC-(-J)) |
| 2 | JIS(ESC-$-@, ESC-(-H)) |
| 3 | JIS(ESC-$-@, ESC-(-B)) |
| 4 | JIS(ESC-$-B, ESC-(-J)) |
| 5 | JIS(ESC-$-B, ESC-(-H)) |
| 6 | JIS(ESC-$-B, ESC-(-B)) |
| 10 | シフト JIS |
| 20 | EUC |

6. dvips について

   dvips は海外で使用されている DVI-PS プログラムである。しかし、このままでは日本語 PS フォントを使用した出力はできないので、 NTT 版 jTeX は日本語 PS フォントを扱える dvips 日本語 Post Script を使用する。アスキー日本語 TeX の DVI ファイルも dvi2dvi を使うことによって出力できる。

# 4 LaTeX2HTML

## 4.1 HTML と LaTeX の関係

HTML とは、WWW (World Wide Web) 文書のハイパーテキストを記述するための言語のことである。これはインターネット上でのホームページの作成にも使用され、これは文書テキストにフォントや文字の大きさなどの指定を書き込むことで文書構造を記述する SGML (Standard Generalized Markup Language) をベースにしている。 SGML は LaTeX などのマークアップ言語をより一般化し、さらに文書を構造化して情報の再利用を可能にしたもので、出力のための組版処理に LaTeX を利用するので両者はとても親和性が高いと言える。 LaTeX は構造をもった文書という概念を備えているし、スタイル指定をドキュメントクラスやドキュメントスタイル、パッケージやスタイルファイルのような形に分離しているので、 HTML との変換も比較的簡単に行うことができる。

## 4.2 LaTeX2HTML の概要と特徴、必要なもの

LaTeX2HTML は Nikos Drakos 氏によって開発され、LaTeX のファイルを HTML 文書形式のファイルに自動的に変換するソフトである。
この特徴として、

1. LaTeX の文書で HTML のリンクが扱える
2. LaTeX の文書中に HTML を直接記述できる
3. 指定に応じて、 1 つの文書を複数の部分に分割できる
4. 条件に応じた文書 (conditional text) が記述できる
5. LaTeX のアクセントや特殊記号を ISO-LATIN-1 文字セットに変換できる
6. ページごとにナビゲーションパネルが作れる。このパネルはカスタマイズもできる
7. 数式、表、図などを PNG 又は GIF ファイルに自動変換し、インラインイメージとして表示したり、リンクとして指定したりできる
8. LaTeX の相互参照をハイパーリンクに展開できるし、脚注や目次、図や表などのリストが扱える
9. ユーザーによる柔軟な設定が可能である

などがある。
7 の PNG や GIF は、インターネットなどで画像データを保存する標準的な形式 (フォーマット) である

例えば、数十ページからなる LaTeX 文書を HTML に変換したとき、 1 つの HTML ファイルにすべてのページを収めると、たいへん大きくて見づらい WWW ページになるが、 LaTeX2HTML では標準で章ごとのページに分割してくれる。そして、次の章を表示するためのボタンなどを各ページに配置したり、目次から直接各章を表示できるようにハイパーリンクを生成してくれる。

LaTeX2HTML では、現在標準的に使われているバージョンの HTML の仕様で記述できないものは GIF イメージに変換して扱う。現在の LaTeX2HTML の場合は変換などのために以下のツールが必要である。

**Perl** LaTeX2HTML 自体は Perl スクリプトなので、 Perl がインストールされていないと使用できない。しかも Perl はバージョン 5.00305 以降が必要で、データベース管理用のモジュールも必要である。

**LaTeX** 数式、図、表などの画像処理には、ソースが LaTeX に組まれるので必要である。

**dvips または dvipsk** dvips, dvipsk は version 5.516 以降が必要である。LaTeX2HTML は、 dvips または dvipsk を使うことが前提に作成されていて、LaTeX の DVI 出力を PostScript ファイルに変換するのに必要である。これら以外の DVI から PS プログラムを利用する場合は修正が必要である。

**netpbm** グラフィック操作を行うユーティリティセットで、クリッピング、回転、減色、背景の透明化、インターレース化、特定形式への変換など、様々なグラフィックフォーマットを変換するためのライブラリである。netpbm は以前 pbmplus と呼ばれていたパッケージに、リリース以降に世界各地で加えられた拡張をまとめて追加したものである。
この netpbm は version 1-Mar-94, または version 9.12 以降必要である。

**Ghostscript** Ghostscript は、バージョン 4.03 以降のものが必要である。この Ghostscript について簡単に説明すると、 L.Peter Deutsch が作成した PS 互換インタープリンタである。

他のユーティリティ LaTeX2HTML にはプリプロセッサをサポートしている機能が備わっているので、インド系言語の変換など、特殊な目的に使うユーティリティも自分の環境に合わせて用意しておく必要がある。

LaTeX2HTML には Web ページに数式を表示するために、

初心者 数学的な内容でない普通の文書に比較的単純な数式が記述

プロフェッショナル インライン形式やディスプレイ形式の数式が記述

エキスパート 数式全体の構造をいくつか小さな部分に分割し処理して記述

という 3 つの数式モードが備わっているが、共通部分は数式を記述するには画像として処理されることである。しかし、 LaTeX2HTML にはこの画像を減らすために、代わりにギリシャ文字や一部の数学記号として特殊なフォントを使えるようにする Unicode が定義してあるが、これは分数などの縦方向に揃えなければ構文に関しては画像が必要になることを付け足す。分数中の積分記号や総和記号といった可変サイズの演算子は画像として処理され、この演算子に添える上付き文字や下付き文字も画像として含まれる。使用方法としてよいのは、エキスパートと Peal コマンドの math 拡張モジュールを使う方法である。これはブラウザで利用可能なフォントの文字を使って多くの数学記号を表現するので必要となる画像の数が少いからである。しかし、現在一般に利用されているブラウザの中には Unicode をサポートしていなかったり、サポートしてあったとしても限定された機能のみしか使えないものやギリシャ文字や数学記号の適切なフォントが備わっていないブラウザもある。

### 4.3 各種言語への対応

TeX では特殊なフォントとマクロパッケージを使用することによって文章が組み込めるが、 LaTeX2HTML にも利用できる。 LaTeX2HTML は、各種の言語用のインターフェースが用意されているが、用意されていない言語でも、別の言語インターフェースを調節し、新しいインターフェースを作成できる。LaTeX2HTML には、英語以外の言語を使って HTML ページを作成するための以下のサポート機能がある。

タイトルとキーワード LaTeX2HTML は HTML ページにタイトルなどの単語を自動的に配置するので、多くの言語はタイトルを翻訳しておかなければいけない

特殊なフォントを利用した画像 LaTeX2HTML では明示的にフォントを切り替えるマクロが識別でき、はっきりと中括弧で区切れば、テキスト内の適切な部分が画像として処理できる

文字セットのエンコーディング 文字セットのエンコーディングについて説明すると英語や日本語以外に使用言語には、

- ドイツ語
- フランス語
- スペイン語

などがあり、他にも地中海言語や中東言語がある。これらは LaTeX2HTML の中には文章中に \selectlanguage コマンドを記述しておくと、テキストの中でどの言語を利用するか指定できる。 LaTeX2HTML の styles\ ディレクトリには、各種のスタイルファイル用 の perl スクリプトの他に

- CJK.perl
- american.perl
- brazil.perl
- english.perl
- german.perl

などの LaTeX $2_\varepsilon$の Babel パッケージで採用されている言語用の perl スクリプトが含まれているが、

> basque(バスク), breton(ブルターニュ), bulgarian(ブルガリア), greek(ギリシャ), iceland(アイスランド), russian(ロシア), swadish(スウェーデン), ukrainian(ウクライナ)

などは Babel パッケージが対応していないので LaTeX2HTML ではサポートしていない。

プリプロセスと画像 インド系の言語や東南アジア諸国などのもっとも複雑な言語の場合、各種の音訳や転写の方式を利用することによって、標準のラテンキーボードから入力可能な文字を使って符号化して表現できる。

### 4.4 LaTeX2HTML の日本語化

LaTeX2HTML はスクリプト言語の perl で書かれていて簡単に修正できることからたくさんの人が個人で LaTeX2HTML の日本語化やカスタマイズに取り組み、安定してきたのが LaTeX2HTML version 97.X 対応の日本語パッチで、その後 98.1 や 99.1 として公開された。LaTeX2HTML の日本語化に、最初取り組まれたのは 北川 勝浩 氏で、日本語化 perl などを使わない形 ( LaTeX2HTML 98.1 の日本語化パッチ) に改良されたのが 武藤 健志 氏である。

ここで LaTeX2HTML version 99.1 を例で挙げると、

- nkf を用いて適切な漢字コード変換を行い、デフォルトでの出力時に JIS(ISO-2022-JP) に変換
- 「無題」「図 1 」「表 1 」「目次」「参考文献」などを日本語に対応
- 日付を日本語式に対応。ドキュメント内の japanese.perl を変更すれば「神無月」などの月名も使用可能
- 対応するスタイルファイルに、 jarticle, jreport, jbook, j-article, j-report, j-book, jslides を追加

などの機能が追加され、これらの版の LaTeX2HTML には、

1. LaTeX のファイルをうまく処理してくれず、 LaTeX2HTML にちゃんと通るように LaTeX の方を書き直さないといけない
2. インストールがうまくできない
3. 画像処理がうまくできない

などの不都合な点があり、その後改良が行われ 99.2$\beta$, 2000, 2K1$\beta$ が公開されている。この 99.2$\beta$ からはインストールがシンプルになり、他の問題も改善されてきている。現在公開されている最新版 latex2html-2002-2-1.tar.gs である。

また日本語パッチもそれに合わせて更新され、日本語化された LaTeX2HTML では、LaTeX 2$\varepsilon$, dvips, Ghostscript などは

**LaTeX 2$\varepsilon$** アスキー版 pLaTeX

**dvips** pTeX に対応したものが必要

**Ghostscript** 日本語対応されたものが必要

であり、更に文字コード変換のために nkf という日本語エンコーディングの変換を行うツールもある。

### 4.5 LaTeX と HTML との相互変換について

LaTeX で作成文書を LaTeX から HTML プログラムで HTML に変換する方法と、HTML で作成した文書を HTML から LaTeX プログラムで LaTeX に変換する方法がある。しかし HTML で作成した文書を HTML から LaTeX プログラムで LaTeX に変換する方法は、HTML で作成した文書を LaTeX に変換しても LaTeX が持っている機能を十分活用した出力結果はなかなかでないので、 LaTeX で作成文書を LaTeX から HTML プログラムで HTML に変換する方法の方が、

1. LaTeX ではユーザーがマクロを書けるので機能の追加だけでなく、LaTeX から HTML の html.sty のように 2 つの環境の差を吸収することもできるのに対して、 HTML ではこのような機能拡張は簡単にはできなし、新しい機能を追加するときは WWW ブラウザ自体を変更しなければいけない。

2. HTML 文書では、論理的な文書構造を明確に指定することができないことや、 CGI を使用している動的なページでは CGI スクリプト中に HTML 文書の各部分が散在するようになっていることから、 HTML 文書から LaTeX で必要とされる論理的な文書構造を抽出して変換するのは難しい。

3. LaTeX では、章や節などの複雑な構造から数少ないファイルで文書を表現することでき、しかも相互参照は 1 つの文書のなかだけで行われるので認識によって処理できる。反対に、 HTML では複数のファイルがハイパーリンクで相互参照している構成で、しかも、文書の内部と外部参照の区別がない。

4. HTML では主として GIF や PNG データが使われるが、ディスプレイ表示での解像度が低く色数も少いデータから出版にも利用できる高解像度、フルカラーのデータに変換することはできない。これに対して LaTeX ではこのような目的に適した PostScript データが標準的に使われ、しかもデータを GIF や PNG 形式に変換する機能も latex2html ではサポートされている。

などの理由から柔軟で適応性が高いといえる。なにより LaTeX2HTML はきわめて優秀なツールなので、高度な記述もできることやオプションを省略して起動するだけでも一定程度以上の出力が得られる。これ以外にも SGML で文書を記述し必要に応じて LaTeX や HTML に変換する方法もあって、 SGML と LaTeX の関係は似ている。

## 5 対応と比較、検討

### 5.1 CJK パッケージと pTeX の比較

LaTeX の表現において、 CJK パッケージを使用しての pTeX ファイルと使用していない日本語化された pTeX ファイルとの差異を述べる。

**TeX ファイル上での違い** pTeX ファイル場合は、

```
\documentclass[a4j]{jarticle}
\begin{document}
\title{タイトル}
\author{名前}
\date{日付}

日本語の文章

\end{document}
```

となり、 CJK パッケージを使用しての pTeX ファイルの場合は、

```
\documentclass[a4paper]{article}
\usepackage{CJK}
\begin{indentfirst}
\begin{document}
\begin{CJK*}[dnp]{JIS}{goth}
\title{タイトル}
\author{名前}
\date{日付}

日本語の文章

\end{CJK*}
\end{document}
```

となる。ファイル内での差異は、

- pTeX の場合は、ドキュメントクラス指定部分を \documentclass[a4j] {jarticle} と指定されており、 CJK パッケージ使用の pTeX の場合は \ documentclass[a4paper]{article} と指定され、用紙サイズの指定が異なっているのと文字入力指定が jarticle から article と置き換えてある。
- 日本語の文書を書き込む場合、 pTeX は jarticle になっているので書きたい場所に書き込めるが、 CJK 使用の pTeX は書き込みたい部分の前後を \beginCJK*[dnp]{JIS}{goth} と \end{CJK*} で囲まなければいけない。
- \usepackage{CJK} の部分を \usepackage[global]{CJK} と定義することができる。この [global] と定義することによって日本語以外の言語定義の中でも日本語を書くことができるようになる。
- タイトルや名前、日付などは pTeX ファイルにも CJK 使用のファイルにも書き込むことができる。CJK 使用のファイルには \beginCJK*[dnp]{JIS}{goth} の部分で {goth} を min などの他のフォントに置き換えることもできる。

表示での違い 1 つ目は、CJK パッケージを使用しての LaTeX の場合は pTeX と比べて、明朝体でもゴシック体でも言えることだが全体的に文書が読みにくい。2 つ目に、「す。」で文章が終わるとき CJK パッケージ使用の場合は、句読点の後に必ず 1 マス空字になる。

### 5.2 欧文アクセントと出力漢字コードの問題

HTML は通常の文字コードの ASCII 部分 (0x21 - 0x7e) に含まれない文字に関して、始めは &#[番号] や &[名前] によって表示していたが、その後各ブラウザが独自の機能として各種の文字集合を用いて書かれた文書に対応し、指定された文字集合の文字を使って文書を表現できるようになった。それが、 HTML ファイルがどの文字集合を使っているかをブラウザに認識させるやり方である。しかし、以前のブラウザは 1 つの表示では 1 つの文字集合しか扱うことができず、日本語のコードと 欧文アクセントなどを一緒に表示することはできなかった。例えば LaTeX の

日本語の文書 ö ;

を HTML 表示すると

日本語の文書 `ö`

と表示され ö が &#246 となって正しく表示されないし、また、欧文の文字集合を使って表示させようとすると「日本語の文書」の部分が正しく表示されない。HTML 4.01 では charset の仕組みは保持したまま、それに含まれない文字を同じ文書に含める場合に [&#番号] あるいは &[名前] ということを番号 (Unicode) で決めた。 Unicode (ISO 10646) は、日本語や中国語なども含むすべての文字を含んだ文字体系のことである。しかし Netscape 6.x, Mozilla 1.x, MS Internet Explorer 4.0 以降の新しいブラウザではバージョンが新しくなり定義され

日本語の文書 `ö`

が表示できる。 LaTeX2HTML での欧文アクセントは、

1. LaTeX に書かれた ASCII 以外の文字はそのまま出力
2. アクセントなどは &#[番号] のように変換して出力
3. 表にないアクセント等の文字は、`-accent_images` が指定されていれば LaTeX+ dvips + netpbm を使って、数式同様 DVI から EPS から GIF のように変換して画像化する
4. `-accent_images` が指定されていなければアクセント記号を取って表示する

などを表示しているが、 2. の形式のものは古いブラウザでは正しく表示できない。しかし、日本語化パッチでは常に 3. または 4. を行い、古い日本語用のブラウザでも欧文アクセントを含む日本語の文書も表示できるようにしている。
また、出力漢字コードに関しては、 LaTeX2HTML ではソースの

```
<META HTTP-EQUIV="Content-type" CONTENT="text/html;charset=ISO-2022-JP>
```

で日本語パッチの定義が表示されおり、オリジナル pTeX と CJK パッケージはファイル内の

```
\begin{CJK*}[dnp]{JIS}{goth}
```

で定義されている。

### 5.3 アクセント問題の実験

LaTeX2HTML の日本語パッチと CJK パッケージは、漢字コード (JIS, SJIS) を指定してあるファイルを latex2hteml 英語版と latex2html 日本語版でコンパイルを行って HTML 表示させる。そしてソース及び、コードの入出力、コンパイル前後の差異などの結果をまとめる。その前に

- pLaTeX ファイルは \documentclass{jarticle} を定義
- CJK パッケージ用のファイルは \usepackage{CJK} を定義

の 2 種類を用意し、それぞれのファイルの漢字コードを

- EUC–JP
- Shift_JIS

としたもの計 4 種類を用意する。なお、 CJK パッケージの LaTeX ファイルの方は、その内部でも使用する漢字コードを

```
EUC  ... \begin{CJK*}[dnp]{JIS}{goth}
SJIS ... \begin{CJK*}[dnp]{SJIS}{goth}
```

のように指定することになっているのでそれはファイルの漢字コードに合わせてある。使用する LaTeX2HTML のコマンドは

- 英語版 latex2html : 2002-2-1
- 日本語版 latex2html

の 2 種類であるの {JIS} 表示の部分である。手順は、 latex2html の英語版 (表では l2h-E ) で CJK の LaTeX ファイル (EUC–JP, Shift–JIS) と pLaTeX ファイル (EUC–JP, Shift–JIS) をコンパイルし、META タグに出力される CHARSET 指定、生成される HTML ファイルの漢字コードを調べる。 latex2html の日本語版 (表では l2h-J ) でも同様に行う。

| コンパイル | ファイル | コード | META | 出力 |
|---|---|---|---|---|
| l2h-E | CJK | EUC | EUC | EUC |
| | | Shift–JIS | SJIS | SJIS |
| | pLaTeX | EUC | 無 | EUC |
| | | Shift–JIS | 無 | SJIS |
| l2h-J | CJK | EUC | EUC | JIS |
| | | Shift–JIS | SJIS | JIS |
| | pLaTeX | EUC | JIS | JIS |
| | | Shift–JIS | JIS | JIS |

結果において META タグは

```
EUC   = <META HTTP-EQUIV="Content-type" CONTENT="text/html;charset=EUC-JP>
SJIS  = <META HTTP-EQUIV="Content-type" CONTENT="text/html;charset=Shift_JIS>
JIS   = <META HTTP-EQUIV="Content-type" CONTENT="text/html;charset=ISO-2022-JP>
無    = <META HTTP-EQUIV="Content-Style-type" CONTENT="text/css>
```

を意味する。

欧文アクセントは下記のアクセントを例として latex2html の英語版と latex2html の日本語版でコンパイルし LaTeX2HTML に表示させる。

õ, ǒ, o̧, ó, ō, ő, ọ, ô, ȯ, o͡o, o̲, ö, ǒ

これらのアクセントが表示されるかどうかを行ってみたところ、

| コンパイル | ファイル | 表示 |
|---|---|---|
| l2h–E | CJK (kan1.tex) | &#245,o,o,&#243,o,o,o,&#244,o,oo,o,&#246,o |
| | CJK (kan1-s.tex) | &#245,o,o,&#243,o,o,o,&#244,o,oo,o,&#246,o |
| | pLaTeX | &#245,o,o,&#243,o,o,o,&#244,o,oo,o,&#246,o |
| l2h–J | CJK (kan1.tex) | o,o,o,o,o,o,o,o,o,oo,o,o,o |
| | CJK (kan1-s.tex) | o,o,o,o,o,o,o,o,o,oo,o,o,o |
| | pLaTeX | o,o,o,o,o,o,o,o,o,oo,o,o,o |

l2h–E は一部分でアクセントが取り除かれた o と &#245 で表示され、l2h–J の場合は、日本語版でコンパイルを行いアクセントが取り除かれた状態 o が表示された。

### 5.4 実験結果についての考察

元々の LaTeX2HTML は EUC-JP や Shift-JIS などの漢字コードを使い分けしているので、 LaTeX ファイルの漢字コードを自動判別しているのではなく、 LaTeX ファイルに指定されている漢字コードを読み取って、 charset 指定を決めている pLaTeX では漢字コード指定とは記述しない。このときは日本語パッチの採用はできないので、

1. 英語版の LaTeX2HTML でのコンパイルに場合、 META タグの状態は CJK の LaTeX の指示に従う。
2. 日本語化されたファイルは CJK パッケージには対応していない
3. META タグのコード指定と HTML ファイルのコードがあっていない場合は、ブラウザによってはうまく表示できないが、 META タグに指定がない場合はブラウザのコードを切りかえることで正しく表示させることができる。又 l2h–J では内部で nkf を使って生成される HTML を強制的に JIS に変換しているので、実験で l2h–J の 出力が JIS となっている。

などがいえる。

charset の設定については、インストールされた環境で決まる。 pLaTeX の場合、実際にファイル内の記述において漢字コードを 2 つも 3 つも使用しないで pLaTeX が EUC 用にインストールされた場合は、 LaTeX ファイルは EUC の漢字コードを使い、 pLaTeX が Shift-JIS 用にインストールされた場合は Shift-JIS の漢字コードを使って LaTeX ファイルを作成する。やり方は

1. latex2html もインストール時に漢字コードの charset を指定できるようする
2. CJK パッケージ使用の場合は、 CJK パッケージの使用目的に添うようにする
3. ユーザー毎のカスタマイズの機能での設定変更等で、指定をできるようにする

の 3 つがある。アクセント問題については、表示させてみて

- 画像化した場合、必ずしも綺麗に表示されない
  ( 文字列の一部のみ画像になるので文字の大きさ、フォントの種類、文字間の間隔がうまく合わない)
- テキストブラウザではもちろん表示されない
- 現在の HTML 4.01 の流儀や LaTeX2HTML の仕組みから外れている

などの問題がでできた。これらの制限や環境設定はいずれ時間とともに解決していくであろう。

## 6 まとめ

LaTeX で日本語を表現するアスキー版 pTeX, NTT 版 jTeX, CJK パッケージの差異と、LaTeX2HTML への対応の現状と比較、検討を行って、 LaTeX2HTML とその日本語化に関する問題点をどう処理すべきかを研究のテーマとして行った。

今回の実験では、英語版の LaTeX2HTML でのコンパイルの場合、 META タグの状態は CJK の LaTeX の指示に従うことや日本語化されたファイルは CJK パッケージには対応していないこと、又 l2h–J では内部で nkf を使って生成される HTML を強制的に JIS に変換しているので、実験で l2h–J の 出力が JIS となっていることや、LaTeX2HTML の欧文アクセントでは &#[番号] のように変換して出力されることと、-accent_images が指定されていなければアクセント記号が取り除かれた状態で表示されることが分かった。

今回の本当の目的は、日本語 LaTeX の環境が他の言語と比べて特殊すぎて取り込まれていないので LaTeX2HTML の本家の方に取り込むことが目的だったが、日本語の表現と LaTeX2HTML への対応の現状と比較までしか進むことしかできなかった。今後の課題として LaTeX2HTML の本家の方に取り込むことが最終目的だが、 LaTeX2HTML の改良や日本語化についてなどが挙げられる。

最後に、本研究を行うにあたって丁寧な御指導、御助言頂きました学情報電子工学科 竹野 茂治 助教授に心から感謝し御礼申し上げます。また、情報電子工学科竹野研究室、 柴信一郎 氏、松本 賢一 氏、宮入 洋介 氏、山岸 寛 氏には本稿執筆に関して御協力頂いたことを心から感謝いたします。

## 参考文献

[1] L.Lanport: “文書処理システム LaTeX”, (アスキー出版, 1990)

[2] L.Lamport: “文書処理システム LaTeX $2_\varepsilon$”, (ピアソン, 1999)

[3] 磯崎秀樹: “ LaTeX 自由自在”, (サイエンス社, 1992)

[4] 奥村晴彦: “ LaTeX $2_\varepsilon$美文書作成入門”, (技術評論社, 1994)

[5] 藤田眞作 : “ LaTeX $2_\varepsilon$階悌”, (株式会社 星雲社)

[6] M.Goossens and S. Rahtz : “ LaTeXweb コンパニオン TeX と HTML\XML の統合”, (株式会社 アスキー, 2001), CHAPTER 3

[7] 小林龍生、安岡孝一、戸村哲、 三上喜貴: “インターネット時代の文字コード”, (共立出版, 2002)

[8] 高山健三 : “ LaTeX $2_\varepsilon$の概要”, UNIX MAGAZINE **10**, no.2, pp132–137 (1995)

[9] 高山健三 : “ NTT jTeX のインストール”, UNIX MAGAZINE **10**, no.8, pp110–119 (1995)

[10] 高山健三 : “ LaTeX から HTML ”, UNIX MAGAZINE **11**, no.3, pp106–113 (1996)

[11] 高山健三 : “ HTML から LaTeX”, UNIX MAGAZINE **11**, no.5, pp107–114 (1996)

[12] 高山健三 : “ LaTeX2HTML ”, UNIX MAGAZINE **17**, no.7, pp117–120 (2002)