日本語 TEX

倉沢 良一
ASCII Corporation

昭和 62 年 8 月
昭和 62 年 12 月
平成元年 10 月

概 要

これは、日本語 TEX の使い方の注意点と、TEX の日本語化にともない変更あるいは追加された機能について解説したドキュメントです。基本的な TEX の使い方については、TEXbook や LATEX: A document Preparation System. をご覧ください。

1 日本語 TEX の概要

日本語 TEX は、オリジナルの TEX と完全なアッパーコンパチブルを保っています。ですから、これまで欧文で作られたファイルはそのまま日本語 TEX にかけられるはずです。現在のバージョンで扱える漢字コードは、シフト JIS および EUC です。ただし、これらはソースファイルにパッチをあてて、make し直さなければなりません。

日本語（2 バイトコードキャラクタ）のハンドリングに関しては、でき得る限り英字（1 バイトコードキャラクタ）と同じように取り扱えるようにしてあります。したがって、特殊な使い方をしない限り英字と同じようにして、原稿ファイルを作成していくことができます。

しかも、カレントフォントとして英字と漢字とを別々に持っているため、英字と漢字の切り換えは特に気にする必要はなく、そのまま混在して使うことができます。

和文と欧文の処理の違いとしてラインブレークのタイミングとそれに関係する禁則処理、さらに和文欧文が混在した場合のスペーシングの処理などがあります。日本語 TEX では、これらの処理を自動的に行っていますが、柔軟性を高めるために、こうした処理に関係するパラメータを自由に再設定できるようになっています。

フォントに関しては、1 フォントで JIS コードの第 1 水準、第 2 水準全てを扱えるように拡張してありますから、英字フォントと同じように漢字フォン

トを扱え、JFM(TFM) ファイルをつくることにより、自由に使用可能なフォントを増やしていくことができます。

DVI ファイルは、SET2 および PUT2 を使って 2 バイトコードを出力しているため、これまでのものとの互換性は保たれています。また DVI ファイルには、JIS コードを用いて出力しています。

2 原稿を書く上での注意点

基本的には、欧文と和文は全く同じようにして扱えます。ただし、日本語特有の処理機能を持たせているため、原稿を書くうえでは次のことに注意してください。

- コントロールシーケンスにも 2 バイトコードキャラクタが使えるようになっています。したがって、コントロールシーケンスに続けて 2 バイトコードの文字を書き並べる場合は、必ず半角スペースやタブ等で間を区切ってください。

- 欧文の場合、改行は単語間のスペースとして取り扱われますが、日本語の場合、原稿内では自由な箇所で改行が行えたほうが便利です。
  そこで日本語 TEX では、1 行の終わりが 1 バイトコードの場合はスペーサーの挿入を行い、2 バイトコードの場合は何も行わないようになっています。

- 2 バイトコードキャラクタと 1 バイトコードキャラクタが連続する場合、自動的に \xkanjiskip に設定されてているグルーを挿入します。ただしこの処理は、どの 1 バイトコードキャラクタとの間で行うか \xspcode によって指定でき、デフォルトでは a–z, A–Z, 0–9 との間で行われるように設定されています。この処理を特定の箇所で禁止したい場合は、そこに \kern0pt、\hskip0pt などを挿入してください。

- 禁則処理は、禁則の対象となるキャラクタの前方あるいは後方にペナルティーを自動的に挿入することで実現しています。このペナルティーの設定は \prebreakpenalty、\postbreakpenalty によって行います。デフォルト値は、kinsoku.tex に記述されています。

- 現在、jtex、jlatex で扱えるフォントは明朝体とゴシック体です。これらのフォントの指定は、jtex, jlatex ともに、\mc、\gt で行えます。これら以外のフォントは、英字フォントと同じように、対象のフォントにあわせて JFM(TFM) を作れば使用できます。

- フォントの切り換えは、英字と漢字は独立して行われます。jlatex では、\large,\small などサイズ変更のコントロールシーケンスによって、連

動してフォントをチェンジするようにしていますが、jtex ではそのような操作は一切行っていませんので注意してください。ただし、指定したフォントが印字されるためには、それらの字体、サイズのフォントがそろっていなければ行えません。jlatex では、次のフォントが存在するものと仮定しています。

> min5, min6, min7, min8, min9, min10, min10 magstephalf, min10 magstep1, min10 magstep2, min10 magstep3, min10 magstep4, min10 magstep5,
> goth5, goth6, goth7, goth8, goth9, goth10, goth10 magstephalf, goth10 magstep1, goth10 magstep2, goth10 magstep3, goth10 magstep4, goth10 magstep5

これらが、きちんと印字されるかどうかはプリンタドライバの責任です。また、これらのフォントにはカーニング・パラメータが設定されています。カーニングを行いたくない場合は、上記のフォントの代りに以下のものを使用してください。こちらのフォントは、カーニングのパラメータが設定されていないことを除いて上記のものと全く同じものです。

> nmin5, nmin6, nmin7, nmin8, nmin9, nmin10, nmin10 magstephalf, nmin10 magstep1, nmin10 magstep2, nmin10 magstep3, nmin10 magstep4, nmin10 magstep5,
> ngoth5, ngoth6, ngoth7, ngoth8, ngoth9, ngoth10, ngoth10 magstephalf, ngoth10 magstep1, ngoth10 magstep2, ngoth10 magstep3, ngoth10 magstep4, ngoth10 magstep5

- 現在のバージョンでは、全角スペースはほかの漢字キャラクタと同じように扱われます。半角スペースのような特別の処理は行っていませんので注意してください。
- コントロールシーケンス名にも全角文字を使用することができますが次の点に注意してください。
  1. 全角の 1 区、2 区、7 区に含まれる文字は、カテゴリーコードの 12 つまり 'Other_character' と同様の扱いになります。したがって、これらの文字は "エスケープ文字 +1 文字" のコントロールシーケンスとしてのみ扱えます。
  2. そのほかの文字に付いては、カテゴリーコードの 11(letter) と同様に扱われます。また、これらの文字は半角文字と混在して使用することができます。

- 半角カナは使えません。原稿内に半角カナのコードが入らないようにしてください。

3 追加されたプリミティブ

日本語 TEX にはつぎのプリミティブが追加されています。

- **\kanjiskip**

  \kanjiskip=⟨dimen⟩

  連続する 2 バイトコード間に自動的に挿入するグルーの値を格納するレジスタです。ただし、この処理は \autospacing、\noautospacing によって行うか行わないかの指定ができます。
  このレジスタの値は、パラグラフの終わりまたは \hbox の最後の時点で取り込まれ、処理されます。したがって、同一パラグラフ内、あるいは \hbox 内で何度か値を変化させたとしても、最後に設定された値によって全て処理されます。

  使　用　例
  `\kanjiskip=10pt plus 1pt minus 1pt`
  と　す　る　と　こ　う　な　り　ま　す。

- **\xkanjiskip**

  \xkanjiskip=⟨dimen⟩

  連続する 2 バイトコードと \xspcode で指定された 1 バイトコードの間に自動的に挿入するグルーの値を格納するレジスタです。ただし、この処理は \autoxspacing、\noautoxspacing によって行うか行わないかの指定ができます。
  \kanjiskip と同様のタイミングで取り込まれ処理されます。

  使用例
  `\xkanjiskip=10pt plus 1pt minus 1pt`
  とすると　alphabet　や　number　との間に挿入されるグルーが　10pt plus 1pt minus 1pt　になります。

- **\sjis**

  \sjis⟨16–bit number⟩

  シフト JIS コードから内部コードへの変換を行います。

使用例
\char\sjis"889F
とすれば、‘亜’ となります。

- **\jis**

  \jis⟨16–bit number⟩

  JIS コードから内部コードへの変換を行います。

  使用例
  \char\jis"3022
  とすれば、‘唖’ となります。

- **\kuten**

  \kuten⟨16–bit number⟩

  区点コードから内部コードへの変換を行います。16 進 4 桁の上 2 桁が区を下 2 桁が点であると解釈します。

  使用例
  \char\kuten"1003
  とすれば、‘娃’ となります。

- **\xspcode**

  \xspcode⟨8–bit number⟩=⟨0|1|2|3⟩

  2 バイトコードと 1 バイトコードの間に自動的にスペースが挿入されますが、このプリミティブによってどの 1 バイトコードとの間でこの処理を行うかどうかを指定できます。この指定は 0–3 のいずれかを選択することにより次のような動作を選択できます。

  – 0 を設定することによりこの 1 バイトコード文字と 2 バイトコード文字の間での処理を禁止します。
  – 1 を設定することでこの文字と直前の 2 バイトコード文字との間にのみスペースを挿入することを許可します。
  – 2 を設定することにより直後の 2 バイトコード文字との間にのみスペースを挿入することを許可します。
  – 3 を設定することで前後の 2 バイトコード文字との間でのスペースの挿入を許可します。

  使用例
  \xspcode‘1=0

\xspcode‘2=3
\xspcode‘;=2
とすることにより、‘1’ に対する処理を禁止し、‘2’ に対して前後の処理を許可します。また、‘;’ に対しては直後へのスペースの挿入を許可します。

- **\inhibitxspcode** 指定した 2 バイトコードとそれに前後する 1 バイトコードの間に自動的に挿入されるスペースを抑制します。この指定は 0–3 のいずれかを選択することにより次のような動作を選択できます。
  - 0 を設定することによりこの 2 バイトコード文字と 1 バイトコード文字の間での処理を禁止します。
  - 1 を設定することでこの文字と直前の 1 バイトコード文字との間にスペースを挿入することを禁止します。
  - 2 を設定することにより直後の 1 バイトコード文字との間にスペースを挿入することを禁止します。
  - 3 を設定することで前後の 1 バイトコード文字との間でのスペースの挿入を許可します。

  このプリミティブは、日本語 TEX のバージョン 1.4 で追加しました。

  使用例
  \xspjcode‘？=0
  \xspjcode‘（=2
  \xspjcode‘）=1

- **\jcharwidowpenalty**

  \jcharwidowpenalty=⟨number⟩

  パラグラフの最後の全角文字が、孤立して改行されるのを防ぐためのペナルティです。またパラグラフの最後が 1 文字以上の 1 区、2 区、7 区に含まれる文字の場合は、その直前にある全角文字に対してこのペナルティが使われます。

- **\autospacing**

  \autospacing

  このプリミティブによって 2 バイトコード間へのグルーの自動挿入を許可します。

- **\noautospacing**

  \noautospacing

このプリミティブによって 2 バイトコード間へのグルーの自動挿入を禁止します。

- **\autoxspacing**

  \autoxspacing

  このプリミティブによって 2 バイトコードと 1 バイトコード間へのグルーの自動挿入を許可します。

- **\noautoxspacing**

  \noautoxspacing

  このプリミティブによって 2 バイトコードと 1 バイトコード間へのグルーの自動挿入を禁止します。

- **\prebreakpenalty**

  \prebreakpenalty⟨16–bit number⟩=⟨number⟩

  指定する文字の前方に挿入するペナルティ値を設定します。この指定は、行頭禁則の指定にあたります。

  使用例
  `\prebreakpenalty‘あ=1000`
  とすることにより、‘あ’ の前方に 1000 のペナルティ値が付けられます。

- **\postbreakpenalty**

  \postbreakpenalty⟨16–bit number⟩=⟨number⟩

  指定する文字の後方に挿入するペナルティ値を設定します。この指定は、行末禁則の指定にあたります。
  ただし、\prebreakpenalty と \postbreakpenalty を同一の文字に対して同時に設定することはできません。同一の文字に対して両方の指定を行った場合、後からの設定が有効になります。

- **\jfont**
  基本的な動作は、“\font” と同じです。ただし、‘\showthe’ などのプリミティブと組み合せた場合にカレントの漢字フォントを返します。フォントの定義は、\font, \jfont のどちらを使っても英字フォント、漢字フォントの定義が行えます。

- **\jfam**
  ‘\jfont’ と同様に “\fam” とほとんど同じ動作をします。‘\showthe’ などとの組み合せることで、カレントの漢字フォントファミリーを返し

ます。ただしこのプリミティブでは、英字フォントのファミリーを漢字ファミリーとして定義できてしまいますので注意してください。

- **zw, zh**
  プリミティブとは異なりますが、**em, ex** と同じように zw、zh は、それぞれカレントの 2 バイトコードフォントの幅（全角幅）および高さの単位記号です。

  使用例
  `\baselineskip=1.5zh`
  `\hsize=20zw`
  でカレントフォントの高さの 1.5 倍の値が \baselineskip に代入され、\hsize が全角 20 文字分に設定されます。

3.1 自動挿入されるスペースについて

\kanjiskip や \xkanjiskip の自動挿入は、単純に連続する文字列ばかりでなく shift_amount が 0 である \hbox 内の文字との前後関係においても行われます。これは、\hbox が入子状になっている場合でも同じです。つまり、その \hbox 内に現れる最初の文字と最後の文字とが、box の前後の文字に連続する文字列として解釈され、そこに指定されたスペースが挿入されることになります。\vbox はその対象とはなりません。具体的には以下の例を参考にしてください。

```
あ\hbox{A}い                                ⟶  あ A い
あ\hbox{\hbox{A}}い                         ⟶  あ A い
あ\hbox{\hbox{\hbox{A}}}い                  ⟶  あ A い
あ\hbox{\hbox{\hbox{\hbox{A}}}}い           ⟶  あ A い
あ\hbox{A\hbox{\vbox{\hbox{A}}}B}い         ⟶ あ AAB い
あ\hbox{\hbox{\vbox{\hbox{A}}}B}い          ⟶  あAB い
あ\hbox{A\hbox{\vbox{\hbox{A}}}}い          ⟶  あ AAい
あ fi い                                    ⟶  あ fi い
あ\hbox{\hbox{\vbox{\hbox{A}}}}い           ⟶  あAい
あ\hbox{\vbox{\hbox{\hbox{A}}}}い           ⟶  あAい
あ\vbox{\hbox{\hbox{\hbox{A}}}}い           ⟶  あAい
A\hbox{あ}A                                 ⟶  A あ A
A\hbox{\hbox{あ}}A                          ⟶  A あ A
A\hbox{\hbox{\hbox{あ}}}A                   ⟶  A あ A
A\hbox{\hbox{\hbox{\hbox{あ}}}}A            ⟶  A あ A
A\hbox{あ\hbox{\vbox{\hbox{あ}}}あ}A        ⟶ A あああ A
A\hbox{\hbox{\vbox{\hbox{あ}}}あ}A          ⟶  Aああ A
A\hbox{あ\hbox{\vbox{\hbox{あ}}}}A          ⟶  A ああA
A\hbox{\hbox{\vbox{\hbox{あ}}}}A            ⟶
AあA                                        ⟶
A\hbox{\vbox{\hbox{\hbox{あ}}}}A            ⟶   AあA
A\vbox{\hbox{\hbox{\hbox{あ}}}}A            ⟶   AあA
```

4　日本語化されていないプリミティブ

次のプリミティブついては、日本語化（2 バイトコードに対応）されていません。

- \catcode, \sfcode, \mathcode, \delcode, \mathchardef は、2 バイトコードに対してそれぞれの定義を行うことはできません。
- \lccode, \uccode は、2 バイトコードをそのままとおし、ローワケース、アッパーケースの変更は行いません。

5　問題点

現在のバージョンの日本語 TEX には、次のような問題点があります。使用の際には注意してください。

- 禁則処理および 1 バイトコードと 2 バイトコード間に挿入するグルーは、その処理の対象となる文字を単に文字コードでのみ判断しています。これは、2 バイトコードの文字に対しては全く問題ないのですが、1 バイトコードの文字で問題が生じます。それは、TEX の文字がコードと一対一に対応していないためで、フォントにより、特定のコードに割り付けられている文字が変ってしまいます。たとえば、‘{’ はフォント cmsy の 66_{16} に対応付けられています。66_{16} は、アスキーコードで ‘n’ を示すわけですから禁則の対象文字とはなっていないはずです。

  したがって、今のままでは ‘{’ のような文字に対して自動的に禁則処理やグルーの自動挿入を行うことは不可能です。