

詰め組用パッケージ

乙部 徹己

2001 年 5 月 20 日

1 詰め組とは

よく知られているように、和文文字は全角を基準として組まれる。ここで、全角基準とは、行末をそろえるというような特殊な調整箇所をのぞきいてはすべての文字が高さ（と深さを加えたもの）に等しい幅を持つ正方形だとしてそれを単に並べるように組むということを意味している。たとえば「日」と「興」では実際の字形部分（グリフ）が持つ幅は同じではないが適宜空白を補って正方形の大きさとして組む。どのようなバランスでどの程度空白を埋めるかはそれぞれの字形のデザイナーが決定することである。和文は伝統的にこのように組まれるので、デザイナーは通常それを仮定して空白などのバランスを決定する。これは現在でも全く変わらない和文組み版の根本原則であり、文字の送り量を全角から変更すると、読みづらくて仕方がないものができるってしまう。

しかしながら、今述べたことはあくまでも本文組である。章や節の題をおくような場合、このような場合には通常は大きな文字が用いられるが、その空白によって締まりのない印象を与えることがある。たとえば、鉤括弧のたぐいは、左右に半角量のアキをあけるべきであるが、一般にタイトルではそのような括弧類が前後に持つ空白を削除した方が締まりがよく見える。同様のことはひらがなやカタカナについてもいえる。小学校のとき、書き方の授業で、ひらがなは漢字よりも小振りに書くように指導されたはずである。その方がバランスよく見える。言い換えれば仮名文字の周囲には大きな空白が空いているので、それを少し削った方が見栄えよく見えることがある。

このようなわけで、和文であっても全角基準で組まない方が適切である箇所が存在する。レベルの高い組み版所では、 \TeX の原稿を処理する前に別のプログラムを使用し、 \section などの命令の引数を読みとり、適宜 \kern を挿入したりしていたようである。しかしながら、適切なツメ量は素人には判断も付かず、手間が増加するので好まれるものではなかった。ところが、Windows 環境において標準で提供される明朝とゴシックである「MS 明朝」と「MS ゴシック」は、実はツメの情報を持っている。この情報を抽出してツメを実現するというのがこのパッケージの目的である。前述したように、ツメの対象は約物（括弧・句読点類）と仮名（ひらがな・かたかな）である。これは MS 明朝フォントでももちろん同様である。漢字部分は詰めないのである¹。

2 ツメ処理できる文字

前述したように、このパッケージによる詰め処理は、対象となるフォントが持っているツメの情報に依存している。従って、一部のフォントに対してのみ使用可能である。従来、本文用和文フォ

¹初めて Windows にツメの情報を持つ和文フォント（プロポーショナル和文フォント）が搭載されたとき、多くのコンピュータ雑誌は「日」と「興」のような例を取り上げて、幅を変えるのが自然だと説いていた。しかしながらこれはもちろん誤りであって、そのようなツメ情報は含まれていない。あくまで仮名と約物・記号類がツメの対象である。

ントはすべて全角基準で組まれることから、実際の出力時に用いるフォントに関しては非常に融通の利くものであった。つまり、存在したのは「和文用のフォントメトリック」であって、「MS 明朝」で出力しても、「リユウミン」で出力しても、あるいは写研の写植機を用いて「石井細明朝」で出力してもかまわなかったし、明朝である必要もなく、「教科書体」「ゴシック体」などで出力してもよかった。どれも共通の配置だったからである。

しかしながら、欧文ではそうはいかない。同じローマン系のフォントであっても「Times New Roman²」と「Computer Modern Roman³」では、同じアルファベットであっても文字幅等が全く異なるために、一方で適切に組んだところに他方の文字を当てると文字が重なったりして見るも無惨な出力となってしまう。

和文詰め組は、たとえば同じ明朝体⁴であっても、欧文のローマン体といった場合の意味と同じく、実際の字形に適した組を行う。そのため、同じ明朝体のフォントであっても異なるデザインのもので代用出力を行うことはできない。あくまでも特定の出力環境で用いられるフォントに限られる。和文の場合、欧文のときの Computer Modern 族に対応するような TeX ユーザーが必ず持っているというようなフォントは残念ながら存在していない。従って、和文詰め組で可搬性を保証するのは不可能である。文書を DVI のレベルで配布するのはほぼ不可能であろう。このパッケージで配布されるものは、以下に限られる。

- 「MS 明朝」「MS ゴシック」に加え、「MS UI Gothic」の Windows 標準フォント。
- RICHIO 社の製品「TrueType World」パッケージに含まれるフォントでツメの情報を持っているもの（いくつかその情報を含まないフォントもある）。

Windows 標準フォントは現在ではもっとも可搬性が高いと思われるが、あくまで Windows ユーザー環境に限られることに注意すること。RICHIO 社の製品は購入したユーザーであれば Macintosh, Unix でも使用可能であろうと思われるが、購入を必要とするため、配布先にそのフォントが存在するとは限らない点に注意を要する。少なくとも現状では DVI レベルで配布することは不可能である⁵。

TrueType World に含まれる「平成明朝体」にはツメ情報が含まれており、そのためのファイルもこのパッケージと共に配布されている。平成明朝体は、財団法人日本規格協会 文字フォント開発・普及委員会によってデザインされたもので JIS 規格ものではあるが、字形として優れているとはされていない。「MS 明朝」などはリョービ社のデザインによるものをやはり RICHIO が作成したものであるが、こちらの方が優れていると思われる。また、平成書体は同じ字形を数多くの企業が作成しているものの、会社によって微妙な差も多く、TrueType World に含まれている RICHIO の平成明朝に従ってツメを行ったものが、他社の平成書体でも正しく出力できるとは限らないようである。

3 パッケージの使用法

パッケージは二つある。tumeegumi.sty と hg.sty である。共に次のようにして使う。

²Monotype 社が Times 紙用に開発したフォント。Adobe 社が PostScript の標準フォントとして採用したのはその Linotype 社版で「Times Roman」と呼ばれる。Times Roman はオリジナルの Times New Roman とやや異なる字形を持つ文字がある。

³やはり Monotype 社が開発した数多くの Modern 族（主な特徴として文字を構成するストロークの太い場所と細い場所の差が大きい）の一つである Monotype Modern 8A というフォントを Knuth が METAFONT を用いて再現したもの。

⁴文字のストロークの最後に存在する三角形上の突起（うろこ）や、シンニョウなどに見られる字形の省略が最大の特徴。

⁵しかしながらこれは当然のことであって、DVI ファイルの形式は、可搬性の全くないものである。現状では可搬性にもっとも富む配布形式は PDF といわざるを得ない。

```

\usepackage{tumegumi}
\usepackage{hg}

```

tumegumi は、標準の明朝とゴシックを「MS 明朝」と「MS ゴシック」とに変更するものである。変更しても実はツメは起こらないので、これは標準のものとほとんど変わらない。しかし大きな文字（`\Large` 以上）を出力しようとする、自動的にツメ処理が行われるようになる。内部的にはツメ処理は、 \LaTeX 2\epsilon の文字属性のうち「シェーブ」としても実装されているのでフォント選択命令で使用することもできるのであるが、そのための命令は `\@tumesshape` と `\text@tume` であって、一般ユーザの手に届かないようになっている。ツメ処理は不用意に本文中で使われるべきではないということである。クラスファイル・パッケージ作者は自由にツメフォントにアクセスできる。またパッケージオプション 13Q を付加すると、10 ポイントに対する和文の大きさが 13 級に変更される。

それに対して hg は RICHIO 社の TrueType World に含まれる数多くの和文書体をしよう可能にするものである。標準の明朝とゴシックは平成明朝と平成ゴシックに置き換えられる。すべての文字は新しいファミリーとして定義されている。以下、ファミリー名 xxx と書いてあれば、`\xxxfamily` という命令と `\textxxx` という命令が定義されているということである。アスタリスク（*）が付加されているものはツメの情報が存在するということである。なお、注意が必要なファミリーとして、丸ゴシックファミリーがある。丸ゴシックファミリーでは、太字のものに対し、標準では「あかね平成丸ゴシック」が用いられている。これをやめたければ、パッケージオプション noakane を用いる。さらに、「平成覚悟疾駆」「平成明朝」「平成丸ゴシック」「プリティフランク」は太字のときに限り、`\textln (\lnshape) \textsd (\sdshape)` によって白抜きと影付きの文字を出力できる（プリティフランクは太字にしなくてもよい）。また、羽衣は `\textlf (\lfseries)` によって細字も可能である。

aps	プリティフランク	ha	羽衣	ppb	創英丸ポップ体 *
bfks	高橋筆楷書体	ke	曲水	hkps	白洲ペン楷書体 *
bpns	千葉字体	lks	小澤楷書体 *	hkjs	白洲極太楷書体 *
bres	高橋隷書体	sngs	祥南行書体 *	hkgs	白洲行草書体 *
cbs	ブーケ *	sks	正楷書体 *	hkfs	白洲太楷書体 *
cres	れいしっく *	sios	創英イオリ書体 *	cks	半古印体 *
kks	教科書体 *	pres	創英プレゼンス EB*	gt	平成角ゴシック体 *
krs	江戸文字勘亭流 *	pna	創英ペン字体 *	mc	平成明朝体 *
cgks	行刻 *	sgus	創英角ゴシック UB*	mg	平成丸ゴシック体 *
gys	行書体 *	ppa	創英角ポップ体 *	snks	有澤楷書体 *

4 PDF ファイル作成について

前述したように、ツメの情報が少しでも使われているなら、その文書をオンラインで配布するとき現状で妥当な解決策は PDF ファイルを作成して配布する以外にはない。配布先で適切な出力を得られることを期待するなら、現実的解決策は、現状では Acrobat（市販品）を用いることになろう。

このとき、たとえ平成明朝を用いているのであろうと、常にフォントを正しく埋め込むことが必須である。また、通常は上記のフォントを PostScript を用いて作業するのは面倒であるし、やや知識を必要とするので、通常は dviout から Distiller プリンタを用いて出力することになろう。Acrobat5

の場合、CJK Screen という設定ではフォントは正しく埋め込まれないようである。「eBook」の設定であれば基本的にすべてのフォントを埋め込んでくれる。もちろん Acrobat に対する若干の知識があれば自分で設定してもよいが、いずれにせよ配布前に Acrobat を起動してフォントの情報を表示し、ツメを行ったフォントが「埋め込みサブセット」と表示されていることを確認してから配布する必要がある。筆者が現在持っている TrueType World は古く、PDF ファイルへの埋め込みが禁止されている。しかしながら最近販売されているものは埋め込みが許可されている。

なお、dviout はデフォルトでは 1 ページごとに出力命令を完結する。これは、Distiller プリンタで出力するときファイルが 1 ページごとに分かれてしまうことを意味する。プリントアウトのダイアログで「Unite Job」を有効にするのを忘れないでほしい。

蛇足ではあるが、PostScript の作成を行いたいユーザーのために TrueType World および Windows 標準フォントについて必要となるフォントの情報を別ファイルとして添付してある (msverbose.txt などのファイル)。これは一般ユーザーには必要のない情報のはずである。

5 JFM について

本節で述べる事柄は、一般ユーザには関係ない。だが、配布されているファイル類をより活用しようとするユーザには重要な情報が記されている。

このパッケージと共に配られているツメでない JFM ファイルは標準的な全角基準の和文用 JFM である。しかしながら注意事項は 10 ポイントのデザインサイズに対して現実に 10 ポイントの出力を与えていることである。min10 などのフォントはデザインサイズは 10 ポイントであるがもう少し小さな出力を与えるようになっている。この理由は簡単である。つまり、「フォントの大きさ」とは一般に文字の高さのことであるが、欧文フォントで 10 ポイントといったとき、本文は一般に小文字であるから、「a・c・e」など、10 ポイントの高さを持たない文字がほとんどであって、大文字や「h・k・l」などの背の高い文字は少数である。そこで、欧文の文字が 10 ポイントといったとき、全体的にはそれよりも小さめの文字が出力される。ところが和文は基本的には 10 ポイントで出力しようするとほとんどすべての文字がその高さを持ってしまう。これを混在させるとバランスがおかしくなってしまう。TeX は和文も欧文もデザインサイズを見てフォント選択を行うので、10 ポイントのデザインサイズを持つ和文に対して本当に 10 ポイントで出力するとあまり美しい体裁とはなくなってしまうのである。そのため、経験則によって min10 はやや小さめのサイズで出力を行うようになっている。

一方 LaTeX 2_ε では 10 ポイントへのユーザーの要求に対して拡大縮小を行ったものを選択させることは容易である。すなわち、今述べたような修正は LaTeX 2_ε レベルで簡単にできるようになっている。これを活用しない手はない。そこで、たとえば欧文 10 ポイントに対して和文 13 級を当てるといったようなことが最近多く行われるようになってきた。min10 はそれに相当するような修正を内部でも行っているので 2 重の調整となり美しくない。しかしながら、たとえば min10 は実体のあるフォントを想定しているわけではないので、ある程度やむを得ぬ処置であろう。ところが本パッケージと共に配布されている JFM はすべて特定のフォントに結びつけられた、実体のあるフォントである。そのようなフォントを対象にする限り、デザインサイズと実サイズに差を設けることには違和感を覚えた。そこで、デザインサイズと実サイズとを同じサイズとして JFM を作成したのである。

これによってもたらされる可能性のある混乱について述べよう。この種の混乱をユーザに生じさせたなら、パッケージデザイナーの責任である。たとえば、本文組はできるだけ互換性を取りたい

とって min10 を用いるような選択を行ったとしよう。そしてツメだけは（ユーザが Windows 環境にいと仮定して）mspmin を用いたとする。この種の混在はユーザに混乱をもたらす。なぜなら、実際に出力される大きさが異なるものとなりうるからである。この混乱を避けるには、注意深くフォント定義を行う必要がある。しかし、ユーザがさらに全体的な拡大縮小を行いたいと思ったとたんこの方式は破綻する。ユーザはフォント定義を注意深く読み直し、拡大率を再計算せねばならない。

そこでこのパッケージでは、標準組のものに対しても、msmin という同じデザインサイズ・実サイズの組のフォントを当てているのである。tumegumi・hg の両パッケージとも、読み込まれると ¥jfont@ratio という値を参照する。デフォルトは ¥jfont@ratio=962 であり、これは 10 ポイントを min10 の実サイズに合わせる。パッケージオプション 13Q はこの値を 925 に変更するものである。