本稿は Linux Japan 誌 2001 年 6 月号に掲載された記事に補筆修正したものです.

斜体のギリシャ文字を使うには

いきなりですが, 次の画面をみてそれぞれのギリシャ文字を表示もしくは印刷する方法がすぐに頭に浮かぶでしょうか？

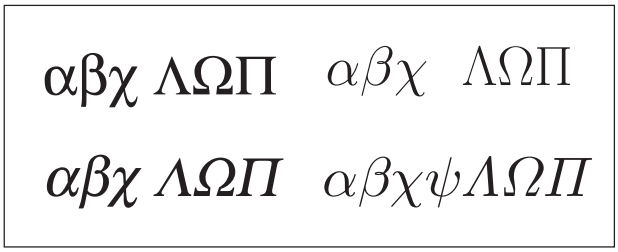

図 1 ギリシャ文字のいろいろ

左上は Tgif で Symbol を選択すると表示されるもので, PostScript の標準欧文書体に含まれています. このフォントは大文字小文字とも立体です. 実は, 筆者はこれに不満があるのです（好みの問題かもしれません）. 右上は, 小文字が斜体で大文字が立体という取り合わせに特徴がありますね. そう TeX の数式モードにおけるギリシャ記号です. 斜体なのはいいのですが, 少し細身で読みづらいという不評を耳にすることもあります. そこで, 筆者の好みの少し太めの斜体が左下にあります. これはその上の立体フォントを斜めにしたものであることが明らかです. tgif では `Ctrl+Alt+t` によりフォントをイタリック (実際は斜めに傾けるオブリーク) にすることができます. ご自分のマシンの tgif で試してみてください. あれれ, でしょう？ そう, 普通にインストールしたままでは何も変化しないのです. 斜めにするには tgif のソースにまで手を加えなければなりません. では, 右下はどうでしょうか？ これは TeX の数式モードでギリシャ記号の大文字を斜体にする `\mit{}` を使うのだ, ということは少し TeX を使い込んだ方には常識です. ところが, これは TeX の出力ではありません. tgif 上で直接描いたものなのです. 年号の記事で, tgif におけるフォントの追加方法を紹介しました. それに従えば, 基本的には TeX の数式イタリックフォント `cmmi10` を Type1 化した `cmmi10.pfb` を使うように設定すればよいのです. が, 事はそう単純ではありません. フォントに細工をしなければならないのです. という訳で, 今回は小賢しい細工の話しでして, なんだつまらんと思う方もいらっしゃるでしょうが, どうぞおつき合いください.

X11 のフォント表示

細工にかかる前に, X のフォントについて簡単におさらいします. JF の XWindow-User-HOWTO のフォントの説明箇所も併せて読んでみてください.

X サーバー

X サーバーはフォントをビットマップ表示します. 一般には X サーバーがクライアントのフォント描画要求に応じてラスタライズします. 現在, X の扱えるフォント形式は, bdf, pcf, speedo, type1 です. さらに FreeType や X-TT を用いることで TrueType フォント (ttf,ttc) を扱えるようになり, 飛躍的に機能が充実しました. X におけるフォントの正式な名称 (XLFD) とフォントファイル名の対応を記述するファイルが `fonts.dir`, `fonts.scale` です. XLFD は繁雑で扱い難いので, 扱いやすい別名との対応を記述するファイルが `fonts.alias` です.

Ghostscript 経由

印刷は Ghostscript を用いて行うのが一般的でしょう. フォントに関しては, ベクトル情報が有効でないデバイス（プリンタ）に対しては GS がビットマップに変換します. X サーバーとはこの点では独立しています. もちろん GS が X に表示する場合は, X11 デバイスに合わせたビットマップを送ります. GS は pcf のような固定サイズのビットマップフォントを扱いません. Type1 を含む PostScript フォント, 欧文 TrueType フォントを扱います. 日本語に関しては, VFlib 組み込むことで扱えるようになっています. PostScript の産みの親 Adobe の立場からは CID フォントを使って欲しいところでしょうが, これは Linux ではなかなか難しく, 最近のバージョン 6.* ににおいての対応が有志の間で行われています [1][W[3]]. GS のフォント設定に関しては, 基本は PostScript におけるフォント名称とフォントファイル名, また別名とフォント名の対応を `Fontmap` に記述することです. また, 日本語に関しては, いろいろな組み込み方がありますが, VFlib を通じてならば `kconfig.ps` が設定ファイルとなります.

Tgif のフォント設定

tgif は X クライアントですから，ディスプレイに表示する場合 X のフォント設定に従います．ただし，印刷は PostScript を用いた高品位なものが望ましいので，GS のフォント設定が決め手になります．X が日本語 TTF を扱えなかった時代には，表示はギザだらけでも印刷時には VFlib(GS の VFlib 化は X よりも先行していました) のおかげで綺麗に印字されたという事態が生じていたのです．

Adobe の Symbol フォントの斜体化

Ghostscript で標準に配布される Type1 フォントは，一般に/usr/share/ghostscript/fonts などにインストールされています．配布系によっては，フリーな欧文フォントが更に追加されています．例えば Plamo2.1 や Vine2.1 では TeX の Computer Modern フォントを Type 1 化したものが追加されているのです．URW++ 社 [2]W[3]から寄贈のあった Adobe の Symbol フォント互換のフォントは `s0500001.pfb` です．これは，筆者の好みに合わない立体ですから，これを素材にして斜体を作ってしまおうと言う訳です．新たに必要な器具は `t1utils`[3]W[3] だけです．簡単にコンパイルできますから，まずインストールしてください．まず，*.pfb はバイナリなので，これをテキストに変換します．さらに，斜体化して別のフォントにするので，名前を変えてコピーしましょう．/ghostscript/fonts/ で作業するには，root の権限が必要です．他の書体の命名規則に従って，斜体を ***20l.pfa(最後は小文字のエルですからお間違えなく) としました．

```
# t1ascii s0500001.pfb s0500001.pfa
# cp s0500001.pfa s0500201.pfa
```

さて，ここで s0500201.pfa を編集します．変更は 2 箇所 23 行目 `/Fontmatrix` の内容を

```
[0.001 0.0 0.0000 0.001 0.0 0.0]
→ [0.001 0.0 0.0002 0.001 0.0 0.0]
```

とします．あと，11 行目の `/FullName` も Oblique を付け加えた方が良いかもしれませんが，どのみち，もう PostScript プリンター非標準フォントなので，PS プリンターでは印字できません．Ghostscript 内部で矛盾しなければ大丈夫です．なぜなら，非 PS プリンタに対しては GS 内部でラスタライズして送り出すからです．他のフォントに合わせて，バイナリ形式 .pfb に戻します．

```
# t1binary s0500201.pfa s0500201.pfb
```

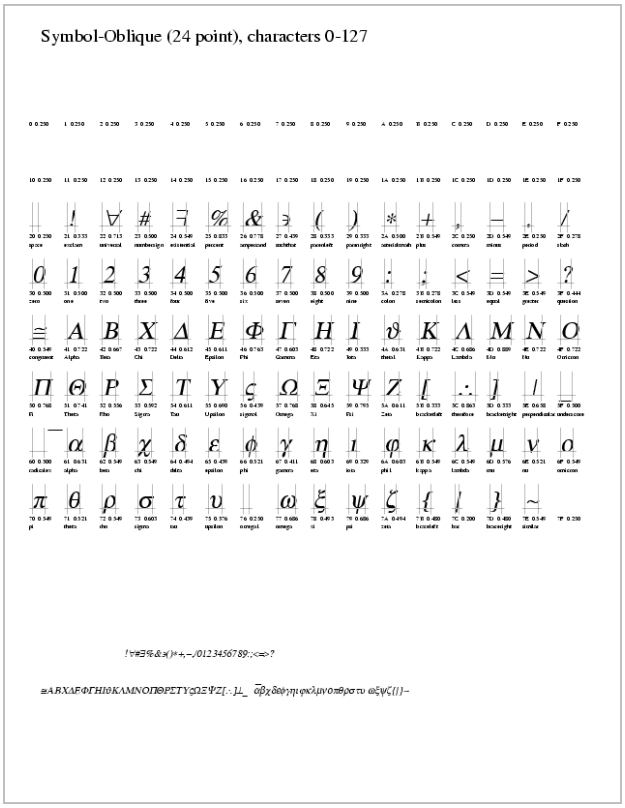

図 2　GS のライブラリにある prfont.ps による /Symbol-Oblique の表示

GS への登録と確認

このフォントを，GS に対し `/Symbol-Oblique` という名前で登録しましょう．`Fontmap` の最後に以下を追記します．

```
/Symbol-Oblique (s0500201.pfb);
```

すると，GS 上で `/Symbol-Oblique` という名前で用いることができます．2000 年 6 月号のこの連載で紹介した，prfont.ps が ghostscript/5.*/ にありますから，確かめてみましょう (図 2)．

```
$ gs prfont.ps
...
GS>/Symbol-Oblique DoFont
...
>>showpage, press <return> to continue<<
```

Enter キーでページを送ります．3 ページ目にリンゴのマークが (^^

X サーバーへの登録・確認

さて，X サーバー自体が表示できるように両方の Symbol フォントを ghostscript/fonts/fonts.dir に登録しましょう．

```
s0500001.pfb （実際は 1 行に記述）
-gs-symbol-medium-r-normal--0-0-0-0-p-0-adobe-fontspecific
s0500201.pfb （同上）
-gs-symbol-medium-o-normal--0-0-0-0-p-0-adobe-fontspecific
```

ここからの作業が，各人の環境で異なりますから，注意してください．まず，このデイレクトリが xtt の設定ファイル config の catalogue に含まれていなくて XF86Config にもフォントパス設定していない（インストールしたままではこの状態）場合には，X のフォントパスに

```
$ xset +fp /usr/share/ghostscript/fonts
（ディレクトリは Plamo2.1 の場合）
```

と加えれば，次のコマンドで表示の可否を確かめられます（図 3）．

```
$ xfontsel -pattern -gs-symbol-*  あるいは
$ xfd -fn -gs-symbol-*
```

もし，XF86Config に

```
  FontPath   "/usr/share/ghostscript/fonts"
```

のようにフォントパスを設定していた場合には，

```
$ xset fp rehash
```

とした後に上記コマンドで表示の可否を確かめてください．

続いて，このディレクトリが X-TT の設定ファイル config の catalogue の項に記されている場合は，一旦 X を終了して X-TT に関する部分を初期化しなければいけません． X サーバーに X-TT が組み込まれている場合（Vine2.1 など）には，単に X を立ち上げ直します．xfs 経由で X-TT を利用するようなシステム（xfs-xtt：Plamo2.1 など）では，xfs(-xtt) を初期化する必要があります．例えば

```
# /etc/rc.d/rc.xfs restart
```

としてから X を立ち上げます．上記コマンドで表示されるかどうか確かめてください．

Tgif に関する設定

これで準備が整いましたので，tgif で使えるようにしましょう．tgif では既に，Symbol が使えるようになっています．そこで，Symbol-Oblique を別フォントとして

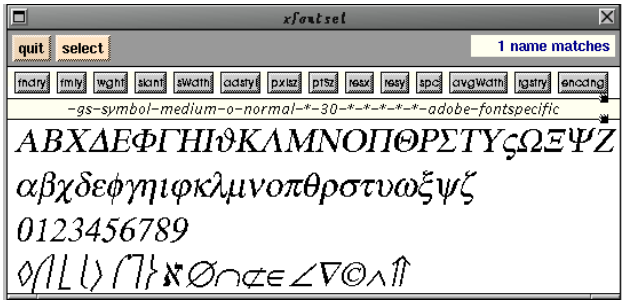

図 3　xfontsel による斜体化 Symbol の表示

```
Tgif.AdditionalFonts:\n\
Symbol-medium-o-normal,adobe-fontspecific, Symbol-Oblique
```

などと追加すれことができます．ただし，メニューには Symbol と表示されてしまい．元からある立体の Symbol と区別しにくいことになります．まあ GS への登録名を変更する（例えば OSymbol）などの工夫を凝らしてこの問題は解決できます．しかし，人間の欲望は限りありません，他の Times（4 つの face が使えるようになっている）などと同等に扱えるようにしたいではありませんか．そこで，tgif のソースに手を加えましょう．ただし，こうなるともう病気かもしれません．

ソースを展開したら，font.c の中で symbol という語句のある場所を捜し出します（emacs なら Ctrl-s ですね）．他のフォントの設定にならって，

```
...
 "symbol-medium-r-normal", "adobe-fontspecific", "Symbol",
 "symbol-medium-r-normal", "adobe-fontspecific", "Symbol",
 "symbol-medium-o-normal", "adobe-fontspecific", "Symbol",
 "symbol-medium-o-normal", "adobe-fontspecific", "Symbol",
...
```

と変更しましょう．これで，Symbol について 2 つの face（立体と斜体）が使えます．「太字はどうするんじゃ？」とつぶやいている貴方，鋭い深い．でも，太字は簡単には得られないのです．

TeX の cmmi フォントを使う

図 1 の右側のフォントは TeX の cmmi10 です．Plamo や Vine では cmmi10 はすでに Type1 化された cmmi10.pfb があって，GS で使うことができます．もしインストールされていない場合には，CTAN[4]W[3] から入手してください．これを素直に tgif の追加フォントに登録するとどうなるかというと，どうもうまくありません．tgif で Symbol フォントが使いやすいのは図 4 のように a のキー入力で α が表示されるというところにあります． cmmi フォントでは a は a のままなのです．もちろん，ギリシャ記号のグリフは cmmi

に含まれていますが，アルファベットを格納する位置とは別の位置に格納されているのですから仕方ありません．無理矢理ギリシャ記号を呼び出すことも tgif では可能なのですが（非アスキー文字は ESC を押してからキー入力すると呼び出せます），Symbol との整合性がとれません．さて，どうしましょう？ そうです，a の位置に α のグリフというように，内容を変えてしまえばよいのです．具体的にはフォントの格納位置が大文字は 65 番から，小文字は 97 番からになっていますから

```
dup 97 /a put   → dup 97 /alpha put
dup 98 /b put   → dup 98 /beta  put
```

のように変更します．これを別名，例えば cmmi10m.pfa という名前で保存して，`t1binary` により pfb 化し，`ghostscript/fonts/` に加えます．また，右に傾いた斜体を左に傾けて疑似的に立体を作成することも可能です．Computer Modern には太字 cmmib もありますからフォントグリフ位置を変換し，さらに立体を作成すれば，全部で 4 つの face が得られます．これらのフォントを GS と X から呼び出せるように ghostscript/5.*/Fontmap と ghostscript/fonts/fonts.dir に登録しましょう．さらに，tgif で使うには Tgif.AdditionalFonts: の項に書体 cmmi10m（もちろん 4 つの face が使えます）として追記します．

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
α	β	χ	δ	ε	ϕ	γ	η	ι	φ	κ	λ	μ	ν	o	π	θ	ρ	σ	τ	υ	ϖ	ω	ξ	ψ	ζ

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	B	X	Δ	E	Φ	Γ	H	I	ϑ	K	Λ	M	N	O	Π	Θ	P	Σ	T	Y	ς	Ω	Ξ	Ψ	Z

図 4　キー入力と Symbol フォントの対応表

ギリシャ語 TTF

かなり苦労しましたが，これでやっと斜体のギリシャ文字が扱えます．筆者がなぜ立体のギリシャ文字に違和感を覚えるのかと自問してみると，物理や数学の教科書にいき当たります．物理では物理量を表す変数は斜体と決っているからです．図 5を見てください．上は斜体を下はやむなく立体を使った場合です．下の図に対して微妙な違和感を覚える方は私と同じくたぶん物理中毒です．たぶんギリシャに暮らす人々が日常使う文字はそうではなく，特に大文字は立体が一般のようです．フリーのギリシャ語 TTF をダウンロードしよう

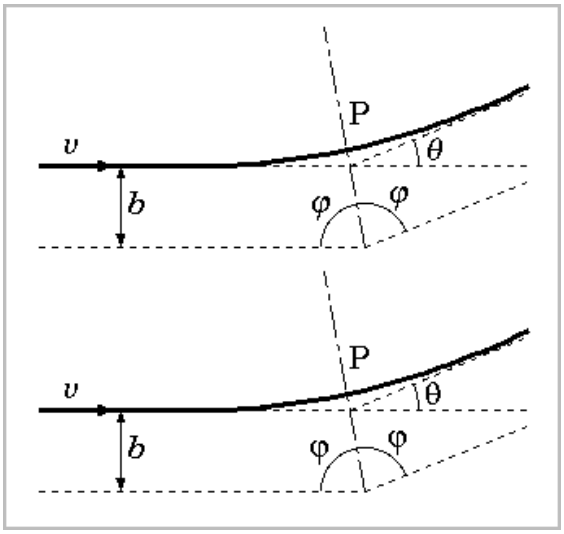

図 5　Symbol フォント：斜体と立体の印象の違い

と思っていろいろなサイト [5]W[3] を調べましたが，結局，圧倒的に立体が多かったです．ちょっと変わったデザインの Korinthus（古代都市コリントなのでしょう）には斜体がありました（図 6）．

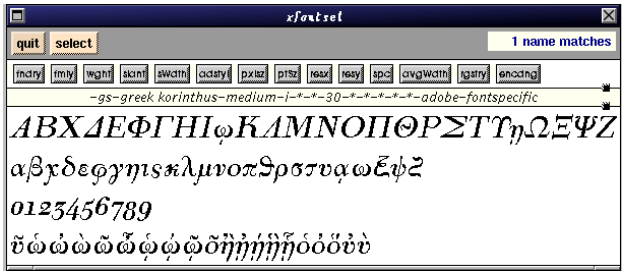

図 6　ギリシャ語 TTF：古代都市コリント風？

参考文献

[1] gs-cjk プロジェクトのページ
`http://www.gyve.org/gs-cjk/index-j.html`

[2] URW 社のホームページ
`http://www.urwpp.de/english/home.html`

[3] T1 ユティリティのサイト
`http://www.lcdf.org/~eddietwo/type/#t1utils`

[4] 毎度お馴染み Ring Sever プロジェクトの CTAN
`ftp://ftp.ring.gr.jp/pub/text/CTAN/fonts/cm/ps-type1/`

[5] 例えば，“Reading, Writing and Printing in Greek”
`http://www.hri.org/fonts/unix/`,

あるいは “Software for Classicists and IT resources”
http://info.ox.ac.uk/departments/classics/software/software.html