本稿は Linux Japan 誌 2000 年 5 月号に掲載された記事に補筆修正したものです.

図面を Tgif で

UNIX における図面のファイル形式は，PostScript が事実上の標準となっています (ビットマップ画像についてではありません，念のため)．この PS ファイルを扱うツールの紹介をしている訳ですが，前回のグラフ作成ツール Gnuplot，Xgraph，GNU Plotutils に続いて，ドロー系の代表で Tgif を取り上げます．Tgif は独自のファイル形式 (拡張子は通常 .obj を用います) をデフォルトとしていますから，PS ファイルを直接編集することができません．そこで，図面用のファイル形式変換ツール Pstoedit の活躍する場ができます.

図面作成ツール：Tgif

Tgif はバージョン 4.0 以降，かなり見栄えが良くなったのと，gettext を採用したおかげて，メニューなどに日本語を表示することができるようになりました．glibc2.* ベースであれば，gettext 機能が組み込まれていますので簡単に利用できます．しかし，libc5 ベースの Plamo 1.* では，自分で gettext を導入する必要があります．そう難しくはないので是非導入しましょうといいたいのですが，自身のない方は glibc2.1 ベースの Plamo 2.0 を待った方がいいかも．この頃，解説記事が多いのでご存知とは思いますが，`/usr/share/locale/ja_JP.ujis/LC_MESSAGES/tgif.mo` (ja_JP.ujis は自分の環境に合わせて読み変えてください) が，メッセージオブジェクトです．含まれる内容は，英語メッセージと日本語メッセージの対応リストなのですが，msgunfmt を使って見ることができます.

```
msgunfmt tgif.mo
```

『フォントはやっぱ英語の方がバランスが良いなあ』と密かに思ったりする筆者なのですが，何はともあれ親しみ易くなりメデタシメデタシです.

さて，PS ファイルの作成方法ですが．Tgif を実際に起動して，図面を書いた後に PS あるいは EPS を選択する方法と，コマンドライン上から以下のように実行する方法があります.

```
tgif -print -ps ***.obj
```

ファイル形式は，この他 EPS (-eps) やビットマップの GIF(-gif)，XPM(-xpm) 等を指定することができます.

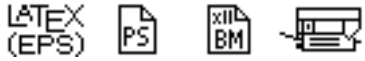

図 1 印刷形式を表すアイコン．左から順に EPS，PS，PS プリンタ，XBM

実際に Tgif を起動した場合を説明しましょう．描いた図面の保存 (Save) は常に Tgif 独自のファイル形式で行われます．それとは別に印刷 (Print) 形式を指定することが可能で，そこで EPS あるいは PS を指定します．パネルウィンドウの 2 段目の一番左が選択された形式を示すアイコンです．クリックして望みの形式に変えてください．デフォルトは 図 1 の左端のように EPS になっていることが多いでしょう.

印刷といってもほとんどがファイルへの出力ですが，図 1 の右端のようなアイコンになっている場合は，(PS) プリンタへの直接印刷となります．その場合には PS ファイルが印刷できる環境に設定をしておかなければなりません．PS プリンタ以外の機種を使っているならば GhostScript を利用することになります．筆者はつぎのような内容のコマンドを /usr/local/bin/gslpr と名付けて使っています.

```
#!/bin/sh
exec gs -q -dNOPAUSE -sDEVICE=mjc360
        -sOutputFile="|lpr " $1 quit.ps
```

もちろん，プリンタ機種に合わせてデバイス名を適切に設定する必要があります．自宅にある EPSON の PM-600C は mjc360(mjc720 も可) でカラー印刷ができます (写真画質は無理のようです)．学校では EPSON の LP-1000 という古いレーザープリンタを使っていますが，epag300 でバッチリです．このように PS ファイルがプリンタに印刷できる状態にしておいて，Tgif のシステムリソースファイル/usr/lib/X11/ja_JP.ujis /app-defaults/Tgif あるいは ~/.Xdefaults に

```
Tgif.PrintCommand:  gslpr %s
```

と記述を加えます．もっとも,『gv でレイアウトを確認してから印刷する癖をつけなさい』と，学生には躾ております.

Canvas 窓だけ

解像度 800×600 のノートパソコン等では，メニューやパネルが邪魔に感じられることがあります．その場合は，Canvas 窓だけ開く (Canvas Window Only) オプションをつけて起動しましょう．ついでに，大きさも指定して,

```
tgif -geometry 500x570 -cwo
```

のようにすると，画面の上下一杯に使えて嬉しいです．作業するには，中ボタンでメニューが呼び出せます．また，右ボタンでモードメニュー (オブジェクトの選択) を呼び出すようにしておきましょう．

```
Tgif.Btn3PopupModeMenu:  on
```

さらに，メニューの一部を押しながらメニュー外側へとドラッグすると，メニューが独立したウィンドウに変身し (WM がタイトルバーを付ける)，常時表示されるようになって便利です．

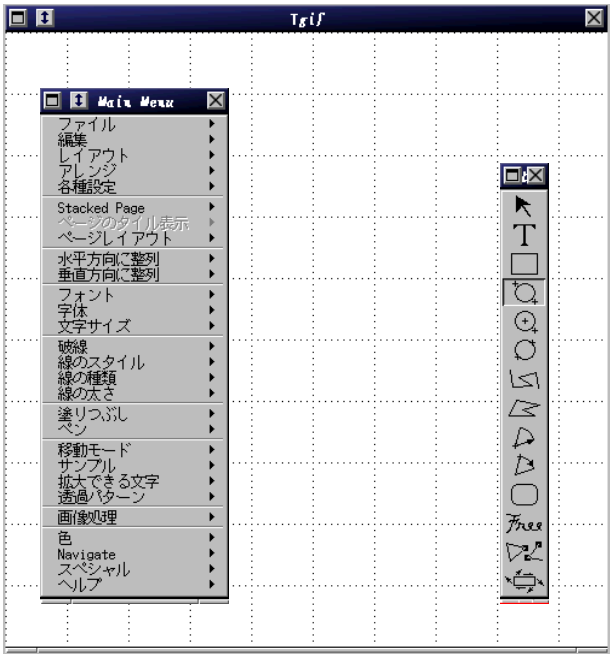

図 2　Canvas 窓のみを開くオプション -cwo を付けて起動した Tgif．中ボタンでメインメニュー (左側) が，右ボタンでモードメニュー (右側) が呼び出されます．

ビットマップの扱い

Tgif はベクター図形だけでなくビットマップも図形オブジェクトの一つとして扱えます．基本的には XPM 形式しか扱いませんから，その他の画像形式はツールで変換して取り込むようになっています．メジャーな GIF，JPEG，TIFF に対しては，リソース指定が予め用意されていて，例えば JPEG に対してならば

```
Tgif.ImportFilter3: JPEG jpg;jpeg \n\
 djpeg -gif -colors 222 %s |giftopnm |ppmtoxpm
```

のような記述があります．一端 222 色の GIF にしてから PNM さらに XPM へと変換していますね．うーん，まわりくどい．Tgif が扱える色数が最大 256 なので，JPEG は減色しなければならないのですが．綺麗に減色するために，このようなまわりくどい変換をしているのでしょう．

Gimp には到底及びませんが，ある程度画像を処理できます．バージョン 4.16 からは screen capture 機能が追加されました．この機能を使うと，Tgif で大きな日本語文字を書いておいて，それをその場ですぐに capture (ファイル→ファイル挿入→スクリーン・キャプチャ) して画像として取り込むことができます．これに画像処理 (編集→画像処理) を施せば，いろいろなロゴが簡単に作れます．

図 3　Tgif の画像処理の実例．一番上は Text オブジェクトなので，画像処理の対象にはならない．二番目は screen capture でビットマップとして取り込んだものに滲み (Blur) をかけたもの．以下，ちらし (Spread)，浮き彫り (Emboss) で処理した．一番下は，輪郭作成を実行した例．

フォントの整備

日本語フォントはリソース

```
Tgif.SquareDoubleByteFonts:
```

のところに XLFD(X Logical Font Description) と PS フォント名を対応つけて記述します．また，欧文フォントも追加が可能で，

```
Tgif.AdditinalFonts:
```

のところに，XLFD の一部とエンコードおよび PS ファイル名の 3 つの項目を記述します．標準配付の tgif.Xdefaults に例がありますからそれにならってください．基本的に，X サーバーが提供する全てのフォント (Bitmap，Type1，TrueType) を使うことが可能です．

なお，欧文のデフォルト，Times，Courier，Helvetica，New Century Schoolbook，Symbol については，Tgif 側でフォント名を限定していますので，勝手な名前のものを使うことができないようです (マニュアルにはできそうに書いてあるのですが，旨くいきませんでした)．

図 4 `xfontsel -pattern -freefont-*` により，freefont パッケージに含まれるフォントのみを表示．画面は caligula という名のフォント．

X11 で見ているだけなら，以上のようにかなりいろいろなフォントが利用できるのですが．印刷するとなると，難しい問題が生じます． Tgif と GhostScript は X11 すなわちディスプレイ上において表示に使用するフォントに元々違いがあるからです．Tgif が使用するフォントは上記のように X クライアントが使う普通のフォントで，**xfontsel** で確かめることができます (図 4)．一方，GS は 日本語に関しては $GS/5.10/kanji /kconfig.ps に，それ以外に関しては $GS/fonts/Fontmap に登録されたフォントを使います (もちろん他のパスを指定することもできます)．日本語フォントは，kconfig.ps が vflib.ps にリンクされている場合には VFlib が使われます．現在 VFlib (/etc/vfontcap で設定) は TrueType を扱うことができますので，結局 Tgif と同じ日本語 TrueType フォントが表示可能になるのです．

欧文の方は，元来 XTT に依らずに同じフォントを使うように設定できます．例えば $GS/fonts にある，Type1(拡張子 pfa, pfb) のフォントを，fonts.dir に登録して，その場で

```
xset +fp /usr[/local]/share/ghostscript/fonts
```

としてフォントのパスに加えた後

```
xset fp rehash
```

と登録を更新すれば GS 附属の Type1 フォントを他の X クライアントで使うことができるようになるからです．最初から X で使えるようにするには，X サーバーの初期設定ファイル /etc[/X11]/XF86Config 等に

```
FontPath /usr[/local]/share/ghostscript/fonts
```

を追加すれば良いことは，他のフォントと同じです．

ところで，GS の扱う論理 PS フォント名とフォントファイル名の対応は/usr[/local]/share/ghostscript /5.10/Fontmap に記されています．どのような形のフォントなのか気になりますね．大小の英数文字を表示させる PS ファイルを作成するシェルスクリプトを以下に示します．

```
#!/bin/sh
cat > $1.ps << EOF
/$1 findfont 25 scalefont setfont
10 750 moveto (ABCDEFGHIJKLMNOPQRSTUVWXYZ) show
10 700 moveto (abcdefghijklmnopqrstuvwxyz) show
10 650 moveto (1234567890) show
showpage
EOF
gv -nocenter -geometry 600x50-0-0 $1.ps
```

スラッシュで始まるフォント名を引数にして

```
showgsfont.sh Times-Roman-Bold-Oblique
```

のように実行しますと，“フォント名.ps” を作成して図 5のように gv で画面表示します．

図 5 showgsfont による GS フォントの表示

調子にのって，色々な GS フォントを表示するスクリプトを作って実行させた結果を図 6に示します．アバンギャルド (AvantGarde)，記号 (Dingbats)，花文字 (古い装飾文字)，スクリプト (手書き風)，平仮名がすでにあるのです．

これらを他の X クライアント (もちろん tgif) でも使えるように，まず $GS/fonts/fonts.dir に登録しましょう．名前は，Fontmap にある PS フォント名に準じて付けるといいでしょう．例えば /AvanGarde-Book ファミリならば，

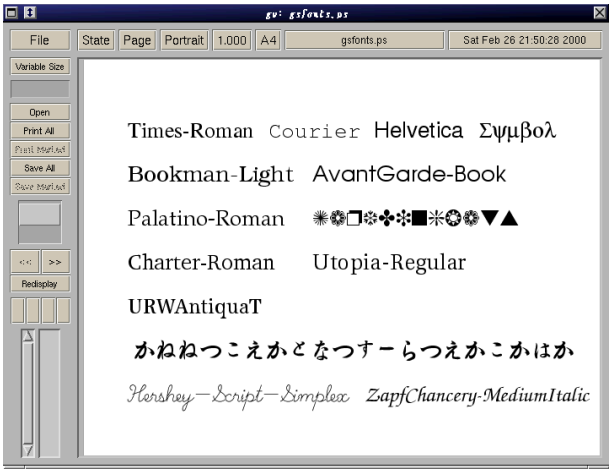

図 6　GS で使われるいろいろなフォント

```
a0100131.pfb -gs-AvantGarde-medium-r-normal--0-0-0-0-p-0-iso8859-1
a0100331.pfb -gs-AvantGarde-medium-o-normal--0-0-0-0-p-0-iso8859-1
a0100151.pfb -gs-AvantGarde-bold-r-normal--0-0-0-0-p-0-iso8859-1
a0100351.pfb -gs-AvantGarde-bold-o-normal--0-0-0-0-p-0-iso8859-1
```

などとします．Tgif 側では，これに対して

```
Tgif.AdditionalFonts:\n\
  AvantGarde-medium-r-normal,iso8859-1,AvantGarde-Book\n\
  AvantGarde-bold-r-normal,iso8859-1,AvantGarde-Demi\n\
  AvantGarde-medium-o-normal,iso8859-1,AvantGarde-BookOblique\n\
  AvantGarde-bold-o-normal,iso8859-1,AvantGarde-DemiOblique
```

のように記述を追加します．他のフォントも同じ要領です．Tgif で作成した PS ファイルを gv で表示させたものと tgif 自身での表示を図 7，図 8に示しますから，比べてください．当り前ですが，同じになりました．

図 7　GS と Tgif 欧米フォントの共通化：GS の表示

実はちょっとこまった問題が起きました．図 6の一番下の左側のフォントは，Hershey と呼ばれる一群

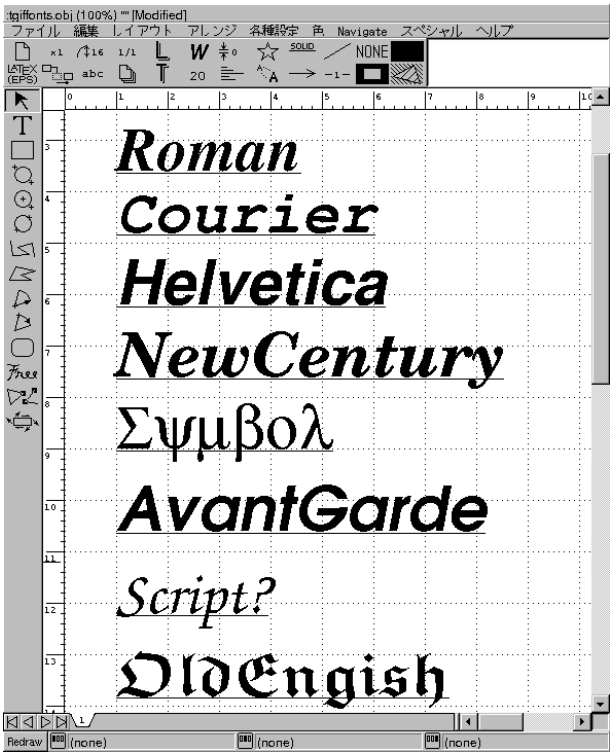

図 8　GS と Tgif 欧米フォントの共通化：Tgif の表示

のフォントに含まれる筆記体なのですが (詳しくは，$GS/5.10/doc/hershey.txt をご覧ください)，これが xfontsel では正常に表示されません．もちろん tgif でもダメです．strings コマンドでフォントの情報を読むと Font-Type1-1.1 などとあり，どうやら他のフォントとはフォーマットが少し違うようです．Hershaey はかなりのフォント (hr で始まるもの) を提供してくれていますので，GS でしか使えないのはちょっと残念です．特に，/Hershey-Gothic(hrg**.pfb) は花文字で，このアンティークなフォントをどうしても使ってみたかったので，工夫してみました．

まず思い付いたのは，TEX の CM フォントを Type1 に変換して使う方法です．Vine ではそのような Type1 CM フォントが収録されています．しかし，惜しいかな，花文字が含まれていません．花文字のフォント名は eufm，eufb のようです．それらを CTAN FTP ミラーサイトから取って来ました．また，ネットワーク上ではフリーな TrueType がころがっていますから (実は ShareWare だったりするので要注意)，見本をみてそれを拾ってくることもできます．CD-ROM 及びフォント見本付きの書籍を購入する手もあります (TrueType と Type1 の両方が含まれている場合が多いです)．ただし，共通化のためには ***.ttf は /usr/lib/X11/fonts/TrueType(あるいは truetype) と

$GS/fonts 両ディレクトリ上で登録しなければなりませんのでちょっと面倒です．

Pstoedit [1]W3

PS ファイルを直接編集できたらどんなに幸せかと思うことがありませんか？．Pstoedit は PS ファイルを種々のベクターグラフィック形式に変換するツールです．Wolfgang Glunz(wglunz@geocities.com) 氏が開発しており，バージョン 3.12 は，Tgif との相性が良くなったようなので簡単に紹介します．

使い方はいたって簡単，

```
pstoedit -f tgif ***.ps ***.obj
```

のようにコマンドを発行します．前回，CGM ファイルビューア gplot の紹介の途中で作成した，car.ps を pstoedit で OBJ ファイルに変換し tgif で編集している様子を図 9 に示します．フォントの変換はほとんどといっていい程ダメなのですが，図形は成功する率が高いようです．45 度回転させ，Script フォントを使って説明を加えてみました．“レイアウト→表示/隠す→メニューバーを表示” のトグル切替でメニューバーを隠してみました．

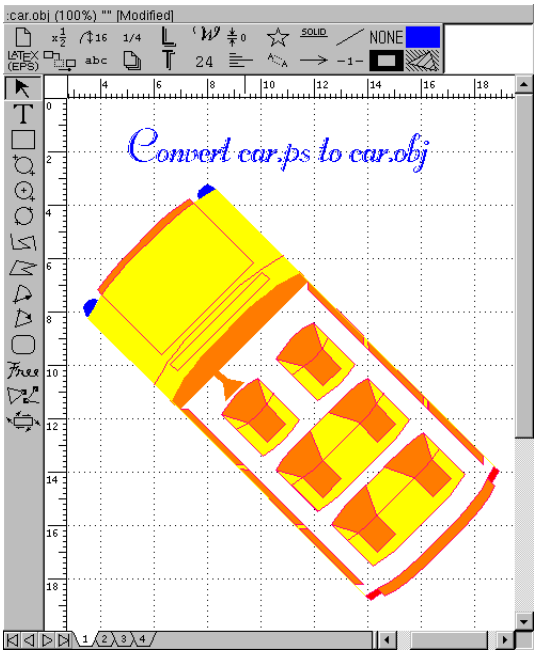

図 9　Pstoedit で Tgif の OBJ 形式に変換した例

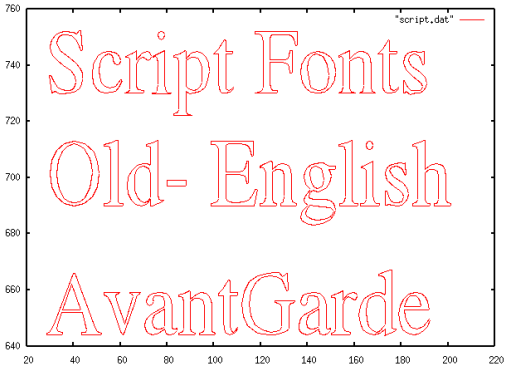

図 10　Pstoedit の変換による，テキストのポリゴン表現

gnuplot 形式？

どのような出力形式がサポートされているかは `--help` オプションで表示されます．そこで面白いものを見付けました gnuplot です．gnuplot の形式の意味が判り難いのですが，マニュアルを読むと納得します．テキストをポリゴンデータで表現するのです．実例を図 10 に示します．フォントが，全て Times-Roman になってしまってますが見事に表現されていますね．人間にはできない力技．筆者はこういうの大好きです．

次回は

Psedit までを終えました．Tgif はまだまだ語り尽くしてない感じです．PS に関連したツールの紹介はまだまだ続きます．こんな終りそうで終らない『巨人の星』のような展開．連載の初期の頃とちっとも変わってませんね (^^;

参考文献

[1] ベクトル形式の図形フォーマットの変換ツール Pstoedit のページ．W3
`http://www.pstoedit.net/pstoedit`