

本稿は [Linux Japan 誌](#) 1999 年 10 月号に掲載された記事に補筆修正したものです。

テキスト主義ばんざい

Linux にもユーザー向けの易しい GUI が必要というご時勢になってます。筆者も X は立ち上げますがその上で CUI, すなわちキーボードを叩いてコマンドを入力するという昔ながらの使い方を愛好しています。その際に無くてはならないのが端末エミュレータです。ファイルの中身を見たりするので (less), 日本語の表示が可能であるものが望ましいです。ところで皆さんご存知の通り, X 上の端末エミュレータには 2 つの流れがあります。X の標準配付 xterm に端を発した color-xterm や kterm の系列と, DEC の VT 端末の名前を冠した xvt を起源とする rxvt (JE では pxvt なんてのもありましたね) や aterm などの系列です。すでに完成していると思われる xterm や kterm に比べて, rxvt はグラフィカルな要素を採り入れる方向で開発が行われています。両者の最も大きな違いの一つは, メニューの呼び出し方式です。xterm 系列はポップアップメニューを採用しており, rxvt 系列は画面上部にメニューバーがあって, プルダウンメニューを採用しています。

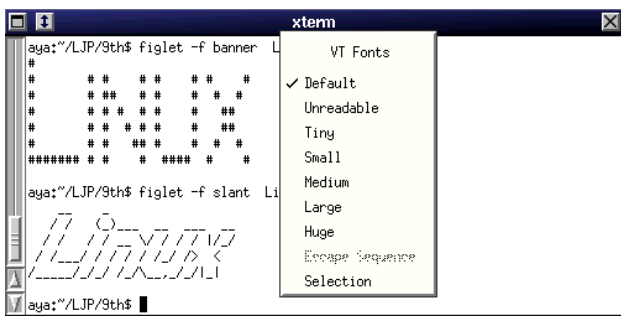


図 1 Xterm 系のポップアップメニュー：Ctrl キーを押しながらマウスボタンをクリックする。3 種類ある。

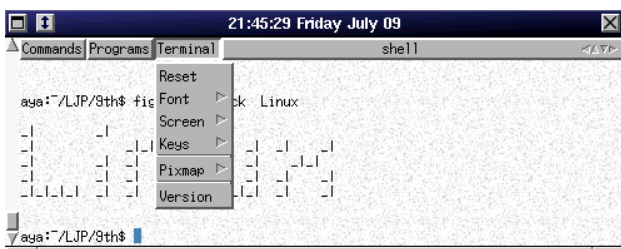


図 2 Rxvt 系のメニューバーとプルダウンメニュー：メニューバーをマウス左ボタンでクリックする。

Rxvt 系

現在も開発の進んでいる rxvt[1][W3] は, カスタマイズ可能なプルダウンメニューと背景 Pixmap が使えて, しかも kterm よりも消費するリソースが少ないといいことづくめのようなので, まずこちらから話を始めることにしましょう。

プルダウンメニューのカスタマイズ

メニューは, 通常ディレクトリ /usr/lib/X11/rxvt に置かれた rxvt.menu で設定されます (メニュー設定ファイル名は, 普通の X クライアントと同様に, /usr/lib/X11/app-defaults/ にある Rxvt の中で指定されます)。

このメニュー設定ファイルはもちろん平テキストですから, ちょこちょこ書き換えをして起動し直したりすれば大体的見当はつきます。しかし, すぐには理解できない謎の文字列があります。メインの rxvt.menu から include される terminal.menu にある

```
/Terminal/*
{Reset}          ^@\Ec
./Font/*
{Normal}         ^@\E]50;#
{Larger}{(M->)} ^@\E]50;#+
{Smaller}{(M-<)} ^@\E]50;#-
```

等の文字列です。まあしかし, 少し考えれば, これらがターミナルの画面制御文字列であり, たぶん /etc/termcap を調べれば定義が理解できる筈と判断できます。実際, \Ec は ibcs ターミナルの定義箇所で c1 (=clear screen) に割当てられています。問題は \E]50;# などです。/etc/termcap にはそのようなエントリはありません。また,

```
infocmp
```

と実行して, 現在使っているターミナルの情報を表示させても, そのようなエントリ見つかりません。ではこの制御はどんな場合に有効なのでしょう。この制御がフォントを変更するものであることは, メニューを実行してみればすぐに判ります。さてよ “フォントの変更” は, 普通のコンソールでは不可能ではないか。そうです, これは X クライアント (複数のフォントが使えますから) に特徴的なものと推測されます。もったいぶらずに最初から結論を述べればよかったですかもしれませんが, 確かに, xterm などの X 上の端末ターミナルのみで働く画面制御なのでした。rxvt のソースを展開してできる doc ディレクトリの xterm.seq に詳

細があります。そこにはテキストパラメタの設定制御

```
\ESC ] Ps;Pt BEL
```

の項があって、細目に

```
Ps=50 -> Set Font to Pt
```

とあります。これで大筋が理解できました。

実は、もう少しだけ不明な点があります。xterm.seq に記述のない制御文字、例えば背景画像の設定などに対応した制御がメニューにあるのです。これは rxvt 独自の拡張部分でして、main.c に記述がありますから

```
grep -8 extension main.c
```

とでもして確認してみてください。表にまとめましたので御覧ください。

表 1 Xterm エスケープ制御とその Rxvt 拡張

Pt : 更新内容	rxvt 拡張
0 : iconName と title	
1 : iconName	
2 : title	
10 : menu	
20 : bg pixmap (背景画像)	
39 : fg (前景色)	
46 : logfile	
49 : bg (背景色)	
50 : font	

また、Breyten Ernsting (bje@dds.nl) および Decklin Foster (decklin@home.com) 両氏による xtermset[2][W³] は、Xterm 系 (もちろん Rxvt 系でも有効です) の画面制御を実行するためのコマンドで、例えば

```
xtermset -geometry 80x15 -T 'date +%X'
```

等として、大きさを 80 桁 15 行に変更し、タイトルに時刻を表示する (1 回だけなのでほとんど無意味ですけど) ことができます。

タイトルバーを書き換える

メニューの書き換えのつもりが、Xterm 系 および Rxvt 系の画面制御の話になってしまいました。そこでお遊びです。ウィンドウ枠の上部に付けられるタイトルが、上記のように起動後も動的に変更できることがわかりましたから、それを使ってみましょう。実は、JE

付属の ~/.bashrc には起動した後に、kterm や rxvt のタイトルを自動変更するスクリプトが記述されています。それを参考にして、次のようなシェルのユーザー関数を ~/.bashrc に定義してみてください (tcsh ユーザーは ~/.tcshrc です)。

```
function nowtime () {
  while true; do
    echo -ne "\033]2;'date'\007" ;
    sleep 1;
  done
}
```

ここで、\033,\007 は 8 進数で表した ESC と BEL として、

```
\ESC ] 2;Pt BEL
```

は表からタイトルを文字列 Pt に変更するコマンドです。そこに、シェルのコマンド展開を用いて、date を実行して得られた文字 (=日付) を代入しています。全体が while 無限ループで囲まれていて、sleep 1 で一秒間隔を置いてタイトルの日付を更新しますから、実質的には時計機能が実現することになります。端末エミュレータを起動後、

```
nowtime &
```

とバックグラウンドで実行させればタイトルバー時計の完成です。

OverTheSpot 漢字入力

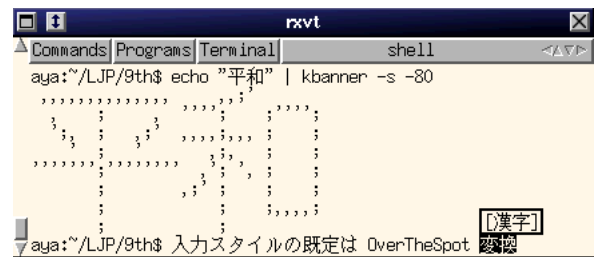


図 3 OverTheSpot 漢字入力方式がサポートされた rxvt の最新バージョン 2.6.0

ところで、雑誌の記事を見ていると kterm の画面はみかけますが、軽いし多機能な rxvt はあまりみかけません。筆者の偏見かもしれませんが rxvt を使うことに躊躇する理由として、日本語変換が Root (別窓を開く) しかサポートされていなかったことが挙げられると思います。しかし、最新のバージョン 2.6.0 では OverTheSpot を含めて全ての方式がサポートされました。起動時にオプション -pt <style> と指定します。

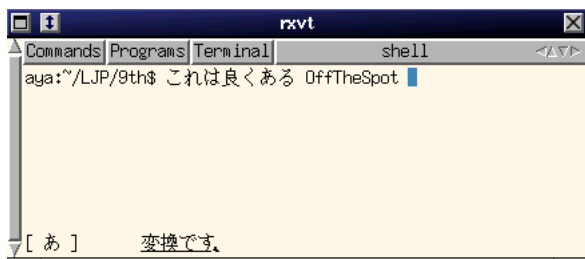


図 4 rxvt 2.6.0 の OffTheSpot 入力方式



図 5 別窓を開く Root : 以前はこの方式のみサポートされてました

style は OverTheSpot, OffTheSpot, Root のいずれかです。一般には kterm と同じく OverTheSpot が最も使いやすいとは思いますが。

先月号で、rxvt は軽いけれど OnTheRoot 変換なので talk などを使いにくいと書いてしまった後に 2.6.0 が公開され、あれーと叫んでしまいました。しかし、これでぐっと使いやすくなったことは紛れもない事実です。

xiterm, eterm, aterm,

rxvt から派生した端末エミュレータが何種類あります。xiterm は国際化されを標榜しています [3][W³]。aterm は AfterStep ユーザー達が rxvt 2.4.8 をベースにして、Alfredo Kojima 氏による背景の透明化を導入して創られました [4][W³]。eterm は一見風変わりなウィンドウマネージャー Enlightenment のプロジェクトによるもので、ライブラリ Imlib を使って画像関係にかなり凝った機能があります。look & feel を指定する theme をサポートしている点に、その哲学が集約されていると良いでしょう [5][W³]。

Rxvt をベースにしているのですが、日本語の表示は問題ないのですが、入力に関しては xiterm, aterm では Root 方式しかサポートされていないです。eterm は、

```
configure --help
```

してみても、Kanji に関する情報が表示されないという寒い状況です。が、Rxvt に準じて、src/feature.h で

```
#define KANJI
```

すれば一応漢字がサポートされます。しかし、kinput2 や vje が動いている場合には、落ちてしまいました(漢字表示は可能です)。

Xterm 系

Xterm 系 に関してはオプション -h により、どのようなオプションがあるかご覧になることをお勧めします。動的に変更可能な画面制御以外の設定に注意を払うとよいのではないのでしょうか。シェルをログインシェルとして起動するオプション -ls (/etc/profile や /.bash_login などを読み込みます)、逆スクロール可能な行数を設定する -sl <num> などは、設定すると使い勝手が向上するかもしれません。また kterm では、行間の空白をドット単位で設定するオプション -lsp <dots> があって、特に、小さい文字を使っている場合には見やすさに大きく影響します。

```
~/VfLib-2.22/tools/
VfLib の動作確認を行なうためのツール群のあるディレクトリ。
文字の一覧表示を行なう ktest。
フォントデータ逆アセンブラ disol。
banner に似たプログラム kban などがある。
```

```
~/VfLib-2.22/tools/
VfLib の動作確認を行なうためのツール群のあるディレクトリ。
文字の一覧表示を行なう ktest。
フォントデータ逆アセンブラ disol。
banner に似たプログラム kban などがある。
```

図 6 kterm における行間の空白調整。既定では 0 ドット(上)で読みにくい、4 ドットにするとかなり見やすい(下)

Rxvt 系 との大きな違いであるメニューに関して、少しだけ。メニューは Ctrl キーを押しながら、マウスボタンを押すと現れます。左ボタンで Main Options, 中ボタンで VT Options, 右ボタンで VT Fonts に関するメニューがポップアップします。これらの内容は /usr/lib/X11/app-defaults/XTerm (KTerm) で設定できます。また kterm なら日本語メニューにもできます。もちろんユーザー毎に ~/.Xdefaults でも可能ですが、あまり意味がないでしょう。以上の事柄は X では常識の範囲ですが、初めての方のため述べました。

Ascii Art

前回、大きな見出しを文字で構成して出力するツールを紹介しました。昔はグラフィカルな表示装置がな

かったので他に方法がなかったのですが、総天然色のグラフィックディスプレイが標準となった現在でも活躍する場所が残っています。前回の X が立ち上がる前のコンソールログイン、原則的には文字しか表示しない電子メールです。電子メールでは MIME によって、もちろん画像ファイルを送ることもできますが、文字だけというきつい制限の中でどのくらいニュアンス伝えられるかが腕のみせどころとなり、むしろアートと考えることができます。そこで文字による絵を作成するツールをいくつか紹介したいと思います。

figlet [6] [W3](#)

メールの最後には気の効いた署名を記したいものです。figlet はそのような要求にぴったりの逸品です。インストール後、showfigfonts を実行すると色々なが表示されます。一部を以下に示します。

```
standard :
  _ _ _ _ _
 / _ _ _ _ \
 | | | | |
 | | | | |
 \ _ _ _ _ /

slant :
  / _ _ _ \
 / _ _ _ \
 ( _ _ )
 / _ _ _ \

lean :
  /
 / /
 / / /
 / / / /

digital :
+++++
|d|l|g|l|t|a|l|
+++++
```

digital なんて泣かせますね。使い方は簡単で、出来上がりを変更するオプション `-f <fontfile>` (指定しない場合は standard が用いられる) くらいを覚えておけば十分でしょう。例えば

```
figlet -f slant Linux
```

のように入力します。残念ですが日本語は使えないようです。

aalib [7] [W3](#)

AA-Project と称する、文字を越えて画像まで手をひらげた試みがあります。グラフィクス端末は日進月歩のビデオカードに依存する能力や機能部分がありますが、テキスト表示はほぼ共通です。そこで、このどんな端末でも動くテキスト表示機能を使ったグラフィックライブラリが創られたという訳です。

図 7 は、aalib に含まれるサンプルプログラム aafire の実行画面です。ちょっと判りにくいかもかもしれませんが、炎がゆらゆらと蠕き、その臨場感に感心させられます。



図 7 aalib に含まれるサンプルプログラム aafire の実行画面

aview

AA-Project から提供された文字による画像ビューアです。画像の濃淡を文字に置き換えて表現します。論より証拠、図 8 をご覧ください。図が複雑になると細部が表現できないので、縦を 50 行使ってしまいました。デフォルトの 80 桁 25 行で表示する場合には、少し遠くから、目を細めてぼやかしてみるのがコツです(^_^;

テキストファイルにするには、ImageMagick の convert など一端画像をネガにします。aview では stdout を出力先に指定して、適当なファイルにリダイレクトすれば出来上がります。

```
convert -negate peng.pnm peng.pgm
aview -driver stdout -height 50 > peng.aww
```

テキスト主義について

冒頭にも申しあげた通り、GUI 全盛の時代に文字端末の話とは、物好きなおっしゃるかもしれません。筆

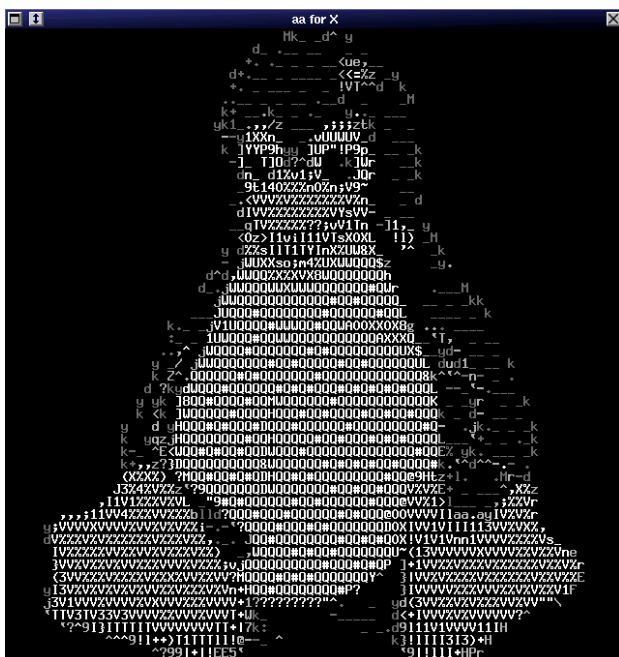


図 8 aview で作成した Linux ペンギンマスコットの AsciiArt : X11 (xterm) への出力例

者は、平素なるべくキーボードから離れないようにしています。キーボードからの入力の方が速いとか、ボタンの組合せで出来ることは限られているからとかいろいろ理由は挙げられます。しかしブラインドタッチの出来ない筆者が主張するのはちと弱いですね。本当の理由は、記憶力の減退してきたオジンにとっては、『コマンドを打ち続けることが唯一『コマンドを忘れない』で済む方法だからと思っています。コマンドを忘れることは、機械に対する主人としての資格を失うことにつながると恐れるからです。

次回は

今回の続きとして、時代に取り遅れた CUI ユーザーここにありという心意気(嘆きと悲しみ?)を込めて、端末エミュレータで動くアプリケーションの数々を紹介することになりますね。ちょっと退屈かもしれませんが、しばしこの懐古趣味におつきあいください。

参考文献

- [1] Rxvt の公式ページ [W³](http://www.rxvt.org/) .
<http://www.rxvt.org/>

- [2] Sourceforge の Xtermset ホーム [W³](http://clts.sourceforge.net/xtermset.html) . Download ができない?
<http://clts.sourceforge.net/xtermset.html>
- [3] Xitem の開発サイトはなくなったようです . ibiblio にソースがあります [W³](http://www.ibiblio.org/pub/Linux/X11/terms/) .
<http://www.ibiblio.org/pub/Linux/X11/terms/>
- [4] Sourceforge の Aterm プロジェクト [W³](http://aterm.sourceforge.net/) .
<http://aterm.sourceforge.net/>
- [5] Eterm [W³](http://www.eterm.org/)
<http://www.eterm.org/>
- [6] Figlet の新しい公式サイト [W³](http://ianchai.50megs.com/figlet.html/)
<http://ianchai.50megs.com/figlet.html/>
- [7] Sourceforge の Aalib プロジェクト [W³](http://aa-project.sourceforge.net/aalib/)
<http://aa-project.sourceforge.net/aalib/>