

本稿は [Linux Japan 誌](#) 2001 年 8 月号に掲載された記事に補筆修正したものです。

Unicode は今

前回, Ghostscript に関連して CID フォントのことを調べていたら、「そういえば Unicode[1][W3] は Linux での程度使えるのか」と気になり始めました。XFree86 が独自に, Xlib の Unicode 化を提案し, 日本を始めとする多バイトフォントの扱いにケリを着けようとして猛烈な反発を受けた事件は記憶に新しいところで [2][W3]。

世界中のあらゆる言語を统一的に記述するために, 数学記号をも含めて文字を一つの表に載せるという発想は自然なものですし方向性としては正しいのでしょう。しかし, 実際に漢字などを常用するわけではない欧米人が中心になって実装するとなると政策・技術両面で問題がでてくるようです。Xlib の場合は Unicode だけで良いという政策が見え隠れして嫌われたようです。技術上も, 完全なフォントファイルができていないという低水準の問題から, 標準ライブラリの問題まで, 全てのアプリケーションが Unicode 対応となるまでにはまだまだ時間がかかりそうです。

記事を書くためにいろいろ調べて, 最後に JF の HOWTO にあたったところ, 筆者のまとめた内容がほとんど全て記述されていたことにショックを受けました。そんなわけで, JF の Unicode-HOWTO[3][W3] というありがたい解説があるので, 今回はそれを補う形で筆をすすめることにしましょう。なお, 符号化という点では, ASCII と互換性がある UTF-8 がもっとも有力となっているようです [4]

フォントファイル

何はともあれ, X 上で字の形がみえなければ話になりません。Unicode の表は [unicode.org](#) から入手できます [1]。CJK Unified Ideographs は, U+E400 より U+9FA5 までに割り当てられています (図 1)。まずは Unicode を表現しうる ISO-10646 のフォントファイルを入手しましょう。上述しましたが, コード表上の全てのグリフが組み込まれたフォントファイルはありません。Unicode-HOWTO に掲載されているフォントの中では, この分野でもっともアクティブな Markus Kuhn (Markus.Kuhn@cl.cam.ac.uk) 氏が提供する

		CJK Unified Ideographs																													
		4E00				4E00								4EDF																	
		4E00	4E01	4E02	4E03	4E04	4E05	4E06	4E07	4E08	4E09	4E0A	4E0B	4E0C	4E0D	4E0E	4E0F	4E10	4E11	4E12	4E13	4E14	4E15	4E16	4E17	4E18	4E19	4E1A	4E1B	4E1C	4E1D
0	一	𠂇	𠂈	𠂉	𠂊	𠂋	𠂌	𠂍	𠂎	𠂏	𠂐	𠂑	𠂒	𠂓	𠂔	𠂕	𠂖	𠂗	𠂘	𠂙	𠂚	𠂛	𠂜	𠂝	𠂞	𠂟	𠂠	𠂡	𠂢	𠂣	𠂤
1	丁	丑	兩	𠂇	𠂈	𠂉	𠂊	𠂋	𠂌	𠂍	𠂎	𠂏	𠂐	𠂑	𠂒	𠂓	𠂔	𠂕	𠂖	𠂗	𠂘	𠂙	𠂚	𠂛	𠂜	𠂝	𠂞	𠂟	𠂠	𠂡	𠂢
2	𠂇	𠂈	𠂉	𠂊	𠂋	𠂌	𠂍	𠂎	𠂏	𠂐	𠂑	𠂒	𠂓	𠂔	𠂕	𠂖	𠂗	𠂘	𠂙	𠂚	𠂛	𠂜	𠂝	𠂞	𠂟	𠂠	𠂡	𠂢	𠂣	𠂤	𠂥
3	七	𠂇	𠂈	𠂉	𠂊	𠂋	𠂌	𠂍	𠂎	𠂏	𠂐	𠂑	𠂒	𠂓	𠂔	𠂕	𠂖	𠂗	𠂘	𠂙	𠂚	𠂛	𠂜	𠂝	𠂞	𠂟	𠂠	𠂡	𠂢	𠂣	𠂤
4	上	且	兩	臨	𠂇	𠂈	𠂉	𠂊	𠂋	𠂌	𠂍	𠂎	𠂏	𠂐	𠂑	𠂒	𠂓	𠂔	𠂕	𠂖	𠂗	𠂘	𠂙	𠂚	𠂛	𠂜	𠂝	𠂞	𠂟	𠂠	𠂡
5	丁	丕	𠂇	𠂈	𠂉	𠂊	𠂋	𠂌	𠂍	𠂎	𠂏	𠂐	𠂑	𠂒	𠂓	𠂔	𠂕	𠂖	𠂗	𠂘	𠂙	𠂚	𠂛	𠂜	𠂝	𠂞	𠂟	𠂠	𠂡	𠂢	𠂣
6	𠂇	世	並	𠂇	𠂈	𠂉	𠂊	𠂋	𠂌	𠂍	𠂎	𠂏	𠂐	𠂑	𠂒	𠂓	𠂔	𠂕	𠂖	𠂗	𠂘	𠂙	𠂚	𠂛	𠂜	𠂝	𠂞	𠂟	𠂠	𠂡	𠂢
7	万	世	喪	𠂇	𠂈	𠂉	𠂊	𠂋	𠂌	𠂍	𠂎	𠂏	𠂐	𠂑	𠂒	𠂓	𠂔	𠂕	𠂖	𠂗	𠂘	𠂙	𠂚	𠂛	𠂜	𠂝	𠂞	𠂟	𠂠	𠂡	𠂢
8	丈	丘	𠂇	丸	𠂇	𠂈	𠂉	𠂊	𠂋	𠂌	𠂍	𠂎	𠂏	𠂐	𠂑	𠂒	𠂓	𠂔	𠂕	𠂖	𠂗	𠂘	𠂙	𠂚	𠂛	𠂜	𠂝	𠂞	𠂟	𠂠	𠂡
9	三	丙	𠂇	丹	𠂇	𠂈	𠂉	𠂊	𠂋	𠂌	𠂍	𠂎	𠂏	𠂐	𠂑	𠂒	𠂓	𠂔	𠂕	𠂖	𠂗	𠂘	𠂙	𠂚	𠂛	𠂜	𠂝	𠂞	𠂟	𠂠	𠂡
A	上	业	个	为	𠂇	𠂈	𠂉	𠂊	𠂋	𠂌	𠂍	𠂎	𠂏	𠂐	𠂑	𠂒	𠂓	𠂔	𠂕	𠂖	𠂗	𠂘	𠂙	𠂚	𠂛	𠂜	𠂝	𠂞	𠂟	𠂠	𠂡
B	下	丛	𠂇	主	𠂇	𠂈	𠂉	𠂊	𠂋	𠂌	𠂍	𠂎	𠂏	𠂐	𠂑	𠂒	𠂓	𠂔	𠂕	𠂖	𠂗	𠂘	𠂙	𠂚	𠂛	𠂜	𠂝	𠂞	𠂟	𠂠	𠂡
C	𠂇	东	𠂇	井	𠂇	𠂈	𠂉	𠂊	𠂋	𠂌	𠂍	𠂎	𠂏	𠂐	𠂑	𠂒	𠂓	𠂔	𠂕	𠂖	𠂗	𠂘	𠂙	𠂚	𠂛	𠂜	𠂝	𠂞	𠂟	𠂠	𠂡
D	不	𠂇	中	𠂇	𠂈	𠂉	𠂊	𠂋	𠂌	𠂍	𠂎	𠂏	𠂐	𠂑	𠂒	𠂓	𠂔	𠂕	𠂖	𠂗	𠂘	𠂙	𠂚	𠂛	𠂜	𠂝	𠂞	𠂟	𠂠	𠂡	𠂢
E	与	𠂇	𠂈	𠂉	𠂊	𠂋	𠂌	𠂍	𠂎	𠂏	𠂐	𠂑	𠂒	𠂓	𠂔	𠂕	𠂖	𠂗	𠂘	𠂙	𠂚	𠂛	𠂜	𠂝	𠂞	𠂟	𠂠	𠂡	𠂢	𠂣	𠂤
F	𠂇	𠂈	𠂉	𠂊	𠂋	𠂌	𠂍	𠂎	𠂏	𠂐	𠂑	𠂒	𠂓	𠂔	𠂕	𠂖	𠂗	𠂘	𠂙	𠂚	𠂛	𠂜	𠂝	𠂞	𠂟	𠂠	𠂡	𠂢	𠂣	𠂤	𠂥

The Unicode Standard 3.0, Copyright © 1991-2000, Unicode, Inc. All rights reserved

631

図 1 Unicode チャートの CJK Unified Ideographs の先頭部分

```
http://www.cl.cam.ac.uk/~mgk25/download/ucs-fonts.tar.gz および ucs-fonts-asian.tar.gz
```

が充実しています。これらを例えば /usr/lib/X11/fonts/ucs で展開して ISO-10646 のフォントだけを残すようにすると管理が楽です。もちろん、XF86Config の FontPath 登録も忘れずに、xfontsel などを確認したいのですが、標準のものではほとんど空白が表示となってしまいます。Mark Leisher (mleisher@crl.nmsu.edu) 氏による ISO-10646 を自動判別するパッチのあつた xfontsel を入手すると便利です [5][W³]

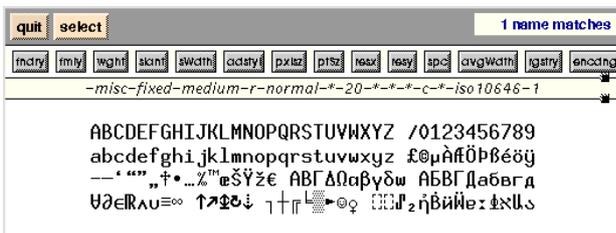


図 2 パッチが当たった xfontsel による ISO-10646 フォントの表示

UTF-8 対応 xterm

CUI 派にとって、次に欲しいのは端末エミュレータでしょう。最近の xterm は UTF-8 をサポートしています。Unicode-HOWTO に書いてある通り

```
http://www.clark.net/pub/dickey/xterm
```

からソースを取得して（バージョンは展開すると判明して 1.56）

```
./configure --enable-wide-char
```

を実行した後すんなり make できます。ロケールを utf-8 にし、さらに英数フォントの倍幅の漢字フォントを使うリソース指定を ~/.Xdefaults などに追記しておきましょう。

```
XTerm*utf8: 1
xterm*VT100*font:\
-misc-fixed-medium-r-normal--18-120-100-100-c-90-iso10646-1
xterm*VT100*wideFont:\
-misc-fixed-medium-r-normal-ja-18-120-100-100-c-180-iso10646-1
```

この例は Kuhn 氏の ucs-font を使う設定です。ucs-fonts.tar.gz を展開するとディレクトリ example の下に UTF-8 のサンプルファイルがあります。図 3 は英語版いろは綴り詞 quickbrown.txt の仮名部分です。ヘブライ文字が下にみえます。

ページャー lv

上記の UTF-8 テキストを眺める際、cat ですと最後まで行ってしまふし今さら more でもないし less を使いたいのですが、UTF-8 を扱うようにするにはパッチが必要ですし、実際に使うには環境変数 LESSCHASET をかえなければならぬなど結構手間です。そこで、強力な多言語ファイルビューア lv を使ってみました（Plamo なら contrib にあります）。UTF-8 ファイルと判っているならば、

```
lv -Au8 quickbrown.txt
```

で自由にスクロールさせて眺めることができます。

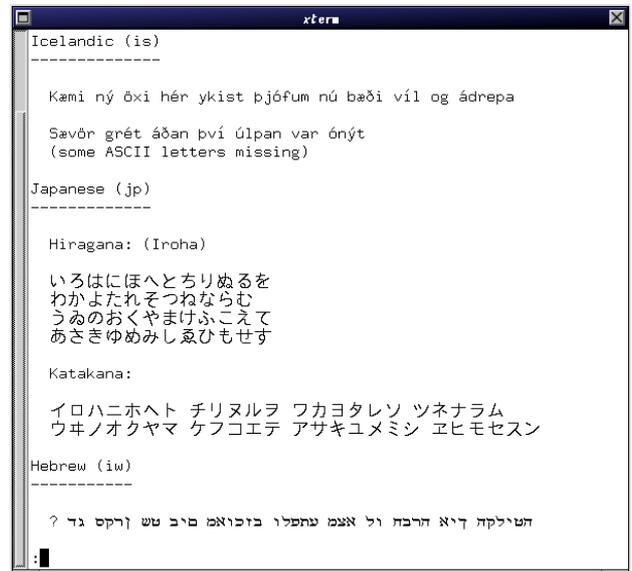


図 3 UTF-8 パッチの当たった xterm の実行画面

エディタ yudit

いよいよエディタです。多言語エディタといえば mule で決まりとは思いますが、ここではその定番をはずして、Gaspar Sinai (gsinai@yudit.org) 氏 [6][W³] による yudit をいじってみましょう。yudit は、X サーバーが提供するフォント以外に、独自に TTF フォントを使うことができます。また、各言語入力も切り替えられるなど、多言語対応としては初心者にも使いやすいものになっています。wananabe-micho.ttf は使えないようなので、次の日本語を含む TTF フォントを入手しましょう。

- <http://ftp.netscape.com/pub/communicator/extras/fonts/windows/> の Cyberbit.ZIP か

CyberCJK.ZIP [W³](#)

- http://www.on.cs.keio.ac.jp/~yasu/jp_fonts.html の東風 (こち) 明朝 [W³](#)

図 4 は, cyberbit.ttf を使って quickbrown.txt を表示しています. 東風明朝よりも Cyberbit の方が多くの言語を含んでいます.

コンパイルとインストールはすんなり終了します. システム全体の設定は

```
/usr/share/yudit/config/yudit.properties
```

で行います. 特別なことは何もないので, フォントをおくディレクトリと名前などをきちんと記述することと, language は ja (ja_JP.eucJP ではありません) を指定する位しか注意すべき事柄はありません. フォントの設定さえしっかりしていれば, 当然ですが, 入力言語を切り替えて次々といろいろな言語を一つの文章中に書くことがすぐにできます. 日本語に関しては kinput2 と kana (漢字変換しない) 入力があります. また, 文字列の方向を切り替えるボタンがあり, これでアラビア文字が入力できるんだろうと妙に感心したりします.

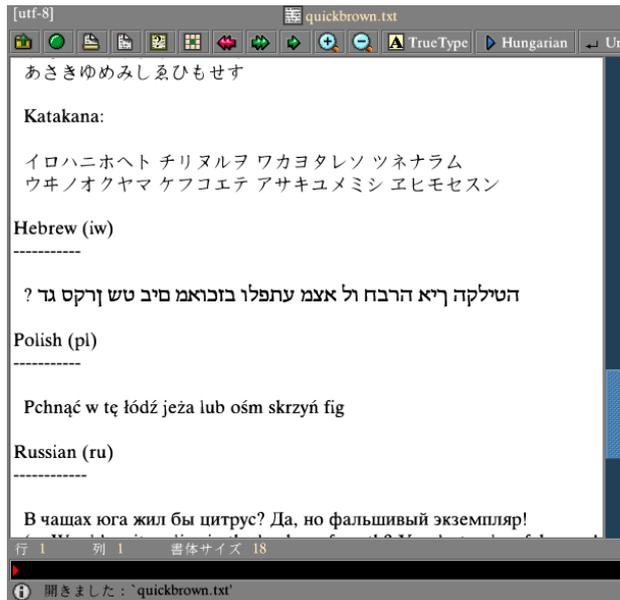


図 4 UTF-8 多言語エディタ yudit の実行例:cyberbit.ttf を用いている

エディタとしての機能は文字入力, カット&ペースト, 検索・置換という最小限のものですが, 印刷に関しては PS に変換し Ghostscript を経由して行い特別な準備が必要ありませんから, すぐに使えます (図 5).

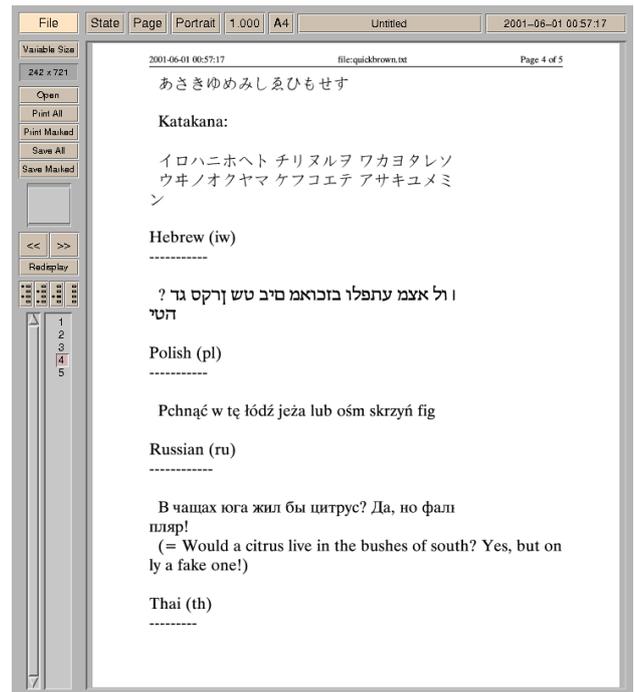


図 5 yudit から印刷プレビューにより gv が起動された例

まとめ

Unicode-HOWTO にはカーネルや tty, あるいは基本コマンドの UTF-8 化方法が記述されていますが, まだ未整備という印象です. ja_JP.UTF-8 でシステム全体が動くにはまだまだ時間がかかりそうです.

参考文献

- [1] <http://www.unicode.org> Unicode といえませんがここ. チャートは /chart ディレクトリにあります.
- [2] <http://www.xfree86.org/pipermail/i18n/> 2000 年 9 月頃からのやりとり.
- [3] <http://www.linux.or.jp/JF/JFdocs/Unicode-HOWTO.txt>
- [4] ‘man 7 utf-8’ その前に ‘man 7 unicode’
- [5] <http://crl.nmsu.edu/~mleisher/download.html>
- [6] <http://www.yudit.org/>