

JBANNER

日本語を含む文字列を拡大し, バナーとして画面に表示する

このドキュメントは, jbanner について記述しています。jbanner は日本語を含む文字列を拡大し, バナーとして画面に表示するコマンドです。CYGWIN の banner と同じ動作になるように作成したつもりですが, ソースを全く参照していませんので, 内部処理は全く異なっているでしょう。また, 細かい動作は異なっていると思われます。

Copyright © 2003 Tadamegu Furukawa

このドキュメントの版数は \$Revision: 1.1 \$ です。

Table of Contents

1	要旨	2
2	機能概要	3
3	動作環境	4
4	インストールとアンインストール方法	5
4.1	インストール	5
4.2	アンインストール	5
5	使用方法	6
5.1	簡単な使いかた	6
5.2	コマンド形式	6
5.3	ヘルプオプション	6
5.4	全体的なオプション	7
5.5	特殊オプション (WIN32)	8
6	蛇足	11
6.1	コンパイル方法	11
6.2	コマンドライン引数の解析	11
6.3	蛇足の蛇足	12
7	取り扱い	13
7.1	著作権、免責等	13
7.2	バグ報告等	13

このドキュメントは, jbanner について記述しています。jbanner は日本語を含む文字列を拡大し, バナーとして画面に表示するコマンドです。CYGWIN の banner と同じ動作になるように作成したつもりですが, ソースを全く参照していませんので, 内部処理は全く異なっているでしょう。また, 細かい動作は異なっていると思われます。

Copyright © 2003 Tadamegu Furukawa

このドキュメントの版数は \$Revision: 1.1 \$ です。

2 機能概要

jbanner は、日本語を含む文字列を拡大し、バナーとして画面に表示するコマンドです。

ある日、メールに日本語のバナーを入れたいと思って、Cygwin の banner を使用したところ、思った通り、日本語は通してくれませんでした。多分、日本語化されたものが有ると思い、インターネット等で検索してみたのですが、違うバナーが沢山ヒットしてしまい、結局目的のものは探し出せませんでした。

「しょうがない。作るか〜。」

「多分、二〜三時間もあれば作れるだろう」

と思って、作り始めました。ま〜、だいたい二時間ぐらいで出来たのですが、どうせなら、オリジナルと同じ動作にしようと思い、オリジナルを操作しながら、動作を真似て作りました。

面倒だったので、オリジナルのソースは見ないで作成したので、細かい部分までは真似しきれないと思います。何れにしろ、オリジナルは多分プログラム内部にフォント情報を持っているでしょう(違うかな?)。欧文のように数が少なければ可能なのですが、日本語は数が多いのでフォントを内部に持つのは得策ではないと判断して、OS からフォント情報を得ています。そんな訳でオリジナルには存在しない機能として、以下の機能があります。

- 日本語を処理できます
- フォントの指定が可能です
- フォントサイズの指定が可能です
- フォントファミリを列挙する事ができます。

フォントの指定が可能になった事により、

```
[498] ~/WORK/jbanner/src: ./jbanner -c 漢 --fontname='@M S 明朝' 漢字
```

```

漢      漢      漢      漢      漢      漢
漢 漢漢漢 漢 漢      漢      漢漢漢      漢漢
漢 漢 漢 漢 漢      漢      漢      漢      漢
漢漢漢 漢 漢 漢 漢 漢      漢      漢      漢
漢 漢 漢 漢 漢漢      漢 漢漢      漢
漢 漢漢漢漢漢漢      漢 漢漢      漢
漢 漢 漢 漢 漢漢      漢 漢      漢 漢 漢
漢漢漢 漢 漢 漢 漢 漢      漢漢漢 漢 漢漢漢漢漢漢
漢 漢 漢 漢 漢      漢      漢      漢      漢漢
漢 漢漢漢 漢 漢      漢      漢      漢      漢
漢      漢      漢      漢      漢      漢
      漢漢漢      漢      漢      漢
漢漢      漢漢漢      漢      漢      漢
漢      漢漢      漢漢漢漢漢      漢漢漢      漢
      漢      漢      漢漢

```

こんな事も出来るようになりました。

3 動作環境

一応, Windows(32bit) 汎用のつもりです。特殊な API は一切使用していませんので, 多分 Windows 95/98/98SE/ME/NT/2000/Xp の何れでも動作すると思います。もしかすると, Win32s でも動作するかもしれません。但し私が動作確認を行った環境は

- Windows 2000
- Windows Xp

のみです。

4 インストールとアンインストール方法

4.1 インストール

インストールは、特殊な事は何も行う必要はありません。‘jbanner.exe’を任意のフォルダ (ディレクトリ) にコピーして下さい。jbanner はコマンドプロンプトから使用するツールですので、環境変数 Pathに含まれているディレクトリにコピーする事をお勧め致します。エクスプローラから、クリックしても何も起こりませんので悪しからず:-p。

4.2 アンインストール

アンインストールは、‘jbanner.exe’を削除するだけです。レジストリは一切触っていませんので、それ以外には何も行う必要はありません。

5 使用方法

5.1 簡単な使いかた

‘jbanner’は Windows のコマンドプロンプト等から使用します。私は Emacs の shell で bash を起動して使用する事が殆どですので、このドキュメントの例も bash のものを使用しているのが殆どです。コマンドプロンプトで使用する場合に於いても特に大きな違いはありませんので、適当に読み替えて下さい。

そのままコマンドプロンプトから ‘jbanner’と入力すると、

```
C:\home\tfuruka1>jbanner
```

```
jbanner: 引数が不十分です。
```

```
Usage: jbanner [-c X] [-w 80] [--usage] [--version] [--help] 表示する文字列
```

といった、簡単な使用方法を表示します。今度は「表示する文字列」を何か入力してみます。ここでは「ほげ」と入力します。

```
C:\home\tfuruka1>jbanner ほげ
```

```

                                X X
                                X X
      X          X          X
      X  XXXXXXXX  X          X
      X          X  X          X
      X          X  X  XXXXXXXX
      X  XXXXXXXX  X          X
      X          X  X          X
      X          X  X          X
      X          X  X          X
      X          X  X          X
      X  XXXXX    X          X
      X  X  XX    X          X
      X  X  X X    X          X
      X  XXXX      XX
```

これが基本的な使用方法です。あとは、オプションで細かい指定がそれなりにできます。

5.2 コマンド形式

‘jbanner’のコマンド形式は以下の通りです。

```
jbanner [options] 表示文字列
```

表示文字列で指定した文字列を拡大して、バナーとして、画面に表示します。オプションは、- または -- から始まる文字又は文字列で、出力形式に対する指示を行う事ができます。オプションの中には、引数を必要とするものもあります。その場合の引数は、コマンドの次の引数として記述する事もできますし、= で連結して指定する事も可能です。例えば、以下の二つはどちらも同じ意味になります。

```
--option argument
```

```
--option=argument
```

因みに jbanner には --option というオプションは存在しません。上記はあくまで例です。

5.3 ヘルプオプション

ヘルプオプションを指定した場合は、jbanner に関連する情報を表示し、直ちにコマンドを終了します。表示文字列を指定した場合も、バナー処理を行いません。

```
-?
```

```
--help
```


`-f=16`

`--fontsize=16`

バナー作成に使用するフォントのフォントサイズを指定します。このサイズはフォントの高さを意味します。フォントの幅は、アスペクトル比から自動算出します。このオプションを指定しなかった場合のフォントサイズは 16 を指定した事になります。フォントサイズに 24 を指定した場合の例を以下に示します (行書体は 16 ドットだと辛いですね)。

[546] ~/WORK/jbanner/src: ./jbanner -c '##' -F=HGS 行書体 -f 24 あ

```

      ##
    ##  ##
  #####  #####
  #####  #####
##  #####
#####
#####
#####      ##
#####  #####
#####      ##  ##
#####      #####  ##
#####      #####  #####
#####  #####  #####
##      #####  #####
##      #####  #####
#####  #####  #####
      #####
#####

```

`--lsfont[=family]`

`family` で指定したフォントファミリのフォント情報を列挙し、コマンドを終了します。フォント名を指定する時に、システムに使用できるフォントを調べる場合に使用する事を想定しています。`family` を指定しなかった場合は、使用可能な全てのフォントを列挙します。フォント情報は、左から

- フォント名
- スタイル名
- 書体名

の順に列挙します。以下に例を示します。

```

[551] ~/WORK/jbanner/src: ./jbanner --lsfont=times
"Times New Roman" "Regular" "Times"
"Times New Roman" "Regular" "Times"
"Times New Roman" "Regular" "Times"
---中略---
"Times New Roman Bold" "Bold" "Times"
"Times New Roman Bold" "Bold" "Times"
"Times New Roman Bold" "Bold" "Times"
---中略---
"Times New Roman Bold Italic" "Bold Italic" "Times"
"Times New Roman Bold Italic" "Bold Italic" "Times"

```

... 以下省略...

```
[552] ~/WORK/jbanner/src: ./jbanner --lsfont="MS ゴシック"  
"MS ゴシック" "標準" "MS ゴシック"
```

jbanner のオプションで指定するのは、フォント名です。フォント名から使用する書体名を得ています。日本語のフォントの場合は、フォント名と書体名は殆ど一致しているようです。

6 蛇足

6.1 コンパイル方法

コンパイルは、VC6.0 と Cygwin の gcc で確認しています。特殊な事は行ってないつもりですので、もっと古いものでも、多分大丈夫だと思います。gcc でコンパイルを通す為に姑息な事 (といっても、昔 MS-DOS の MSC-3 あたりを使用していた頃もこんな事をしていましたが) をしています。早い話が、日本語を正しく処理できない分部の対応をしているだけです。もう少し詳しくいうと、エスケープコード 0x5c が、SHIFT-JIS での漢字コードの 2 バイト目と一致してしまい、特定の漢字コード (2 バイト目がエスケープコードと一致するコード) が正しく処理されないのです。例えば、「表」(0x955c) がそれにあたります。

```
printf("表示\n")
```

だとエラーになりますので、

```
printf("\x95\x5c 示\n")
```

と記述しています。

一応、私がコンパイルを確認したコンパイラを以下に列挙します。

- Microsoft (R) 32-bit C/C++ Optimizing Compiler Version 12.00.8804 for 80x86
- gcc (GCC) 3.2 20020818 (prerelease)

コンパイルは、nmake または make 一発で行えるように考慮したつもりですが、環境によっては、makefile を書き換える必要があるかもしれません (と、いうより、私と全く同じ環境の筈がありませんので、何らかの書き換えは必要になります)。修正方法は、makefile 内に記述していますので、そちらを参照して下さい。

6.2 コマンドライン引数の解析

恥ずかしながら、今回デバッグをしていて始めて判ったのですが、gcc でコンパイルしたものと、MSVC でコンパイルしたもので、コマンドライン引数の処理が異なるようです。Windows のコマンドプロンプトから、以下のコマンドを入力して、その引数の渡し方を調べてみました。

- 'jbanner --fontname='Times New Roman' hoge'

1. MSVC でコンパイルした場合

```
ARGV[0]=<jbanner>
ARGV[1]=<--fontname='Times>
ARGV[2]=<New>
ARGV[3]=<Roman'>
ARGV[4]=<hoge>
```

Times New Roman を一つの引数として渡すつもりで、' で括ったのですが、見事に引数が三つに分割されてしまいました。

2. gcc でコンパイルした場合

```
ARGV[0]=<jbanner>
ARGV[1]=<--fontname=Times New Roman>
ARGV[2]=<hoge>
```

こちらは、意図した通りです。

- 'jbanner --fontname="Times New Roman" hoge'

今度は、' ではなく、" で括ってみました。

1. MSVC でコンパイルした場合

```
ARGV[0]=<jbanner>
ARGV[1]=<--fontname=Times New Roman>
```

```
ARGV[2]=<hoge>
```

意図した通りです。

2. gcc でコンパイルした場合

```
ARGV[0]=<./jbanner>
```

```
ARGV[1]=<--fontname=Times New Roman>
```

```
ARGV[2]=<hoge>
```

意図した通りです。

といった結果になりました。私は殆ど `bash` を使用していますので、シェルが処理してくれるので、気が付きませんでした。

6.3 蛇足の蛇足

一部、デバッグ用に Syslog 出力しています。Syslogd を起動している方は、ご注意 (べつに悪さは与えないと思いますが) ください。

7 取り扱い

7.1 著作権、免責等

本プログラムはフリーソフトウェアです。本プログラムを使用して生じたいかなる結果に対しても作者は責任を負わないこととします。個人の責任に於いて使用して下さい。入手したアーカイブのままの形式であれば、再頒布、転載は可能とします。常識的に扱ってください。

7.2 バグ報告等

バグ報告、改善案 (あるいは、自分で改善した)、質問等ありましたら、是非作者へ「励ましのお便り」(^_^;を送って下さい。特にご自分で改善なさった場合は、私も恩恵に預かりたいので、是非教えて下さい。改善案のご連絡を下さっても、仕事に忙殺されている関係上、ご希望に添える事は出来ないかもしれませんが、バグ修正はなるべく行うつもりです。自分の名誉の為に...(^_^;

メールアドレスは、以下の通りです。

T.Furukawa tfuruka1 at nifty dot com