

Tipos de letra y FreeBSD

Resumen

Este documento contiene una descripción de los diversos archivos de tipos de letra que pueden utilizarse con FreeBSD y el controlador syscons, X11, Ghostscript y Groff. También incluye manuales paso a paso para cambiar la pantalla de syscons al modo 80x60 y para usar tipos de letra de Tipo 1 con los programas anteriores.

Tabla de contenidos

1. Introducción	1
2. Terminología básica.....	1
3. ¿Qué formatos de tipo puedo usar?	2
4. Configuración de una consola virtual en modo de línea 80x60	3
5. Uso de tipos Type 1 con X11.....	3
6. Uso de tipos Type 1 con Ghostscript.....	7
7. Uso de tipos Type 1 con Groff	7
8. Conversión de tipos TrueType a un formato groff/PostScript para groff.....	10
9. ¿Se pueden usar los tipos TrueType con otros programas?.....	12
10. ¿Dónde se pueden obtener tipos adicionales?	12
11. Preguntas adicionales.....	13

1. Introducción

Hay muchos orígenes (sources) de tipos disponibles y a veces no es fácil discernir cómo usarlos en FreeBSD. La respuesta puede estar en la documentación del componente que desea utilizar, aunque puede llevar su tiempo dar con ella; este tutorial intenta proporcionar una solución a quienes puedan estar en esas circunstancias.

2. Terminología básica

Hay muchos formatos de tipos diferentes y sufijos de archivos de tipos asociados. En este artículo abordaremos unos cuantos:

.pfa, .pfb

Tipos Tipo 1 PostScript® . pfa es el formato Ascii y el .pfb es el formato *Binario*.

.afm

Métricas del tipo asociadas al tipo Tipo 1.

.pfm

Métricas de impresión del tipo asociadas al tipo Tipo 1.

.ttf

Tipo TrueType®

.fot

Una referencia indirecta a un tipo TrueType (no es un tipo real)

.fon, .fnt

Tipos de pantalla de mapa de bits

El archivo .fot se usan en Windows® como una especie de enlace simbólico al archivo de tipo TrueType® (.ttf). Los archivos de tipo .fon también se usan en Windows. No conozco ninguna manera de usar este formato de tipo en FreeBSD.

3. ¿Qué formatos de tipo puedo usar?

Qué tipo se puede usar depende de la aplicación. FreeBSD por sí mismo no utiliza tipos. Las aplicaciones y/o los controladores pueden utilizar archivos de tipo. A continuación se muestra una pequeña referencia cruzada de la aplicación/controlador para los sufijos de tipo:

Controlador

vt

.hex

syscons

.fnt

Aplicación

Ghostscript

.pfa, .pfb, .ttf

X11

.pfa, .pfb

Groff

.pfa, .afm

Povray

.ttf

La extensión .fnt es bastante común. Sospecho que la mayoría de las veces en las que alguien quería crear un archivo de tipo especializado para su aplicación elegían esta extensión. Por lo tanto es probable que no todos los archivos que incluyen esta extensión tengan el mismo formato; en concreto los archivos .fnt que usa syscons en FreeBSD pueden no tener el mismo formato que un archivo .fnt en MS-DOS®/Windows®. No he intentado utilizar otros archivos .fnt que no sean los

suministrados con FreeBSD.

4. Configuración de una consola virtual en modo de línea 80x60

En primer lugar se debe cargar un tipo 8x8. El archivo `/etc/rc.conf` debe tener la línea (cambie el nombre del tipo por el que sea más apropiado en su región):

```
font8x8="iso-8x8"      # tipo 8x8 de /usr/shared/syscons/fonts/* (o NO para cargar el
                        tipo por defecto).
```

El comando para cambiar el modo es [vidcontrol\(1\)](#):

```
% vidcontrol VGA_80x60
```

Varios programas screen-oriented como [vi\(1\)](#) pueden determinar el tamaño de la pantalla mediante una llamada de `ioctl` al controlador de la consola (por ejemplo [syscons\(4\)](#)) que determinará correctamente las nuevas dimensiones de la pantalla.

Para hacerlo más sencillo puede añadir estos comandos a los scripts de inicio del sistema para que se ejecuten en el inicio. Agregue esta línea en el archivo `/etc/rc.conf`.

```
allscreens_flags="VGA_80x60"    # Establecer el modo vidcontrol para todas las
pantallas virtuales
```

Referencias: [rc.conf\(5\)](#), [vidcontrol\(1\)](#).

5. Uso de tipos Type 1 con X11

X11 puede utilizar tanto el formato `.pfa` como el formato `.pfb`. Los tipos para X11 se encuentran en varios subdirectorios bajo `/usr/X11R6/lib/X11/fonts`. Cada tipo es una referencia cruzada entre su nombre X11 y el contenido del archivo `fonts.dir` en cada directorio.

Ya existe un directorio llamado `Type1`. La forma más sencilla de añadir un nuevo tipo es ponerla en ese directorio. Una forma aun mejor sería colocar todos los tipos que quiera añadir en un directorio separado y utilizar un enlace simbólico a los tipos adicionales. Esto permite identificar los tipos sin mezclarlos con los originales. Por ejemplo:

```
% mkdir -p /usr/local/shared/fonts/type1
% cd /usr/local/shared/fonts/type1
```

Puede incluir también los archivos README, así como la documentación de los tipos

```
% cp /cdrom/fonts/atm/showboat/showboat.afm .
```

```
% echo showboat - InfoMagic CICA, Dec 1994, /fonts/atm/showboat >>INDEX
```

-bitstream-character-medium-r-normal-xxx-0-0-0-p-0-iso8859-1
\ \
\ \ \ \ \ \ +----+ juego de caracteres
\ \ \ \ \ \ +- ancho promedio
\ \ \ \ \ \ +- espaciado
\ \ \ \ \ \ +- resolución vertical.
\ \ \ \ \ \ +- resolución horizontal.
\ \ \ \ \ \ +- puntos
\ \ \ \ \ \ +- píxeles
\ \ \
tipo familia densidad inclinación anchura estilo adicional

4

```
% strings showboat.pfb | more
%!FontType1-1.0: Showboat 001.001
%%CreationDate: 1/15/91 5:16:03 PM
%%VMusage: 1024 45747
% Generated by Fontographer 3.1
% Showboat
  1991 by David Rakowski.  Alle Rechte Vorbehalten.
FontDirectory/Showboat known{/Showboat findfont dup/UniqueID known{dup
/UniqueID get 4962377 eq exch/FontType get 1 eq and}{pop false}ifelse
{save true}{false}ifelse}{false}ifelse
12 dict begin
/FontInfo 9 dict dup begin
  /version (001.001) readonly def
  /FullName (Showboat) readonly def
  /FamilyName (Showboat) readonly def
  /Weight (Medium) readonly def
  /ItalicAngle 0 def
  /isFixedPitch false def
  /UnderlinePosition -106 def
  /UnderlineThickness 16 def
  /Notice (Showboat
    1991 by David Rakowski.  Alle Rechte Vorbehalten.) readonly def
end readonly def
/FontName /Showboat def
--stdin--
```

Basándonos esta información podríamos usar un nombre como este:

```
-type1-Showboat-medium-r-normal-decorative-0-0-0-0-p-0-iso8859-1
```

Los componentes de nuestro nombre son:

Tipo

Vamos a nombrar todos los tipos nuevos como `type1`.

Familia

El nombre del tipo.

Densidad

Normal, negrita, media, seminegrita, etc. En la salida del comando `strings(1)` que acabamos de mostrar vemos que este tipo tiene una densidad *media*.

Inclinación

roman, cursiva, oblicua, etc. Dado que *ItalicAngle* es cero, se utilizará *roman*.

Anchura

Normal, ancha, condensada, extendida, etc. Hasta que pueda ser examinada, suponemos que

será *normal*.

Estilo adicional

Generalmente se omite, pero esto indicará que el tipo contiene mayúsculas decorativas.

Espaciado

proporcional o monoespaciado. La opción *Proportional* se usa cuando *isFixedPitch* es false.

Todos estos nombres son arbitrarios, pero uno debe tratar de ser compatible con las convenciones existentes. El nombre hace referencia al tipo con posibles comodines del programa X11, por lo que el nombre elegido debe tener algún sentido. Simplemente puede comenzar a usar

```
···-normal-r-normal-···-p-···
```

como nombre, y luego usar [xfontsel\(1\)](#) para examinarla y ajustar el nombre en función de la apariencia del tipo.

Para completar nuestro ejemplo:

```
Haga que el tipo esté accesible para X11
% cd /usr/X11R6/lib/X11/fonts/Type1
% ln -s /usr/local/shared/fonts/type1/showboat.pfb .

Edite fonts.dir y fonts.scale, agregando la línea que describe el tipo
e incrementando el número de tipos que se encuentran en la primera línea.
% ex fonts.dir
:1p
25
:1c
26
.
:$a
showboat.pfb -type1-showboat-medium-r-normal-decorative-0-0-0-0-p-0-iso8859-1
.
:wq

fonts.scale parece ser idéntico a fonts.dir...
% cp fonts.dir fonts.scale

Indique a X11 que las cosas han cambiado
% xset fp rehash

Examine el nuevo tipo
% xfontsel -pattern -type1-*
```

Referencias: [xfontsel\(1\)](#), [xset\(1\)](#), The X Windows System in a Nutshell, [O'Reilly & Associates](#).

6. Uso de tipos Type 1 con Ghostscript

Ghostscript hace referencia a un tipo a través de su archivo Fontmap. Para modificarlo hay que proceder de forma parecida a cuando modificamos el archivo fonts.dir de X11. Ghostscript puede usar los formatos .pfa y .pfb. A continuación ofrecemos una guía paso a paso en la que usaremos el tipo anterior para mostrar cómo usarla con Ghostscript:

```
Coloque el tipo en el directorio de tipos de Ghostscript
% cd /usr/local/shared/ghostscript/fonts
% ln -s /usr/local/shared/fonts/type1/showboat.pfb .

Edite el archivo Fontmap para que Ghostscript esté al corriente del tipo
% cd /usr/local/shared/ghostscript/4.01
% ex Fontmap
:$a
/Showboat      (showboat.pfb) ; % From CICA /fonts/atm/showboat
.
:wq

Use Ghostscript para examinar el tipo
% gs prfont.ps
Aladdin Ghostscript 4.01 (1996-7-10)
Copyright (C) 1996 Aladdin Enterprises, Menlo Park, CA. All rights
reserved.
This software comes with NO WARRANTY: see the file PUBLIC for details.
Loading Times-Roman font from /usr/local/shared/ghostscript/fonts/tir____.pfb...
/1899520 581354 1300084 13826 0 done.
GS>Showboat DoFont
Loading Showboat font from /usr/local/shared/ghostscript/fonts/showboat.pfb...
1939688 565415 1300084 16901 0 done.
>>showpage, press <return> to continue<<
>>showpage, press <return> to continue<<
>>showpage, press <return> to continue<<
GS>quit
```

Referencias: consulte el archivo fonts.txt en la distribución 4.01 de Ghostscript

7. Uso de tipos Type 1 con Groff

Ahora que el nuevo tipo puede ser utilizada tanto por X11 como por Ghostscript ¿cómo se puede usar el nuevo tipo con Groff? En primer lugar y dado que estamos utilizando tipos PostScript® type 1, el dispositivo Groff que vamos a usar es *ps*. Se debe crear un archivo de tipo para cada tipo que queramos usar con Groff. Un nombre de tipo para Groff es simplemente un archivo en el directorio /usr/shared/groff_font/devps. Siguiendo con nuestro ejemplo, el archivo del tipo sería /usr/shared/groff_font/devps/SHOWBOAT. El archivo debe crearse utilizando las herramientas proporcionadas por Groff.

La primera herramienta es **afmtodit**. No está instalada por defecto, pero puede encontrarla en la distribución original. Descubrí que tenía que cambiar la primera línea del archivo, así que procedí del siguiente modo:

```
% cp /usr/src/gnu/usr.bin/groff/afmtodit/afmtodit.pl /tmp
% ex /tmp/afmtodit.pl
:1c
#!/usr/bin/perl -P-
.
:wq
```

Esta herramienta creará el archivo de tipo Groff a partir del archivo de métricas (sufijo .afm). Siguiendo con nuestro ejemplo:

```
Muchos archivos .afm están en formato Mac, con ^M delimitando las líneas
Tenemos que convertirlos al estilo UNIX que delimita las líneas con ^J
% cd /tmp
% cat /usr/local/shared/fonts/type1/showboat.afm |
    tr '\015' '\012' >showboat.afm

Ahora cree el archivo de tipo groff
% cd /usr/shared/groff_font/devps
% /tmp/afmtodit.pl -d DESC -e text.enc /tmp/showboat.afm generate/textmap SHOWBOAT
```

Ahora se puede hacer referencia al tipo con el nombre SHOWBOAT.

Si se usa Ghostscript con las impresoras del sistema no es necesario hacer nada más. Sin embargo si las impresoras usan PostScript® el tipo se debe descargar a la impresora para poder usarse (a menos que la impresora tenga el tipo showboat incorporado o pueda acceder a una unidad en la que esté.) El último paso es crear un tipo descargable. La herramienta **pfbtops** se utiliza para crear el formato de tipo .pfa y el archivo download se modifica para hacer referencia al nuevo tipo. El archivo download debe hacer referencia al nombre interno del tipo. Esto se puede determinar fácilmente a partir del archivo de tipo de groff como vemos a continuación:

```
Cree el archivo de tipo .pfa
% pfbtops /usr/local/shared/fonts/type1/showboat.pfb >showboat.pfa
```

Por supuesto, si el archivo .pfa ya existe, simplemente cree un enlace simbólico para referenciarlo.


```

Obtener el nombre interno del tipo
% fgrep internalname SHOWBOAT
internalname Showboat
Indique a groff que el tipo debe ser descargado

% ex download
:$a
Showboat      showboat.pfa
.
:wq

```

Para probar el tipo:

```

% cd /tmp

% cat >example.t <<EOF
.sp 5
.ps 16
This is an example of the Showboat font:
.br
.ps 48
.vs (\n(.s+2)p
.sp
.ft SHOWBOAT
ABCDEFGHI
.br
JKLMNOPQR
.br
STUVWXYZ
.sp
.ps 16
.vs (\n(.s+2)p
.fp 5 SHOWBOAT
.ft R
To use it for the first letter of a paragraph, it will look like:
.sp 50p
\s(48\f5H\s0\fRere is the first sentence of a paragraph that uses the
showboat font as its first letter.
Additional vertical space must be used to allow room for the larger
letter.
EOF
% groff -Tps example.t >example.ps

Para utilizar ghostscript/ghostview
% ghostview example.ps

Para imprimir
% lpr -Ppostscript example.ps

```

Referencias: [/usr/src/gnu/usr.bin/groff/afmtodit/afmtodit.man](#), [groff_font\(5\)](#), [groff_char\(7\)](#), [pfbtops\(1\)](#).

8. Conversión de tipos TrueType a un formato groff/PostScript para groff

Esto puede llevar un poco de trabajo por la sencilla razón de que depende de algunas herramientas que no se instalan como parte del sistema base:

ttf2pf

Herramientas de conversión de TrueType a PostScript. Esto permite la conversión de tipos TrueType a archivos de métrica de tipo ascii (.afm).

Disponible en <http://sunsite.icm.edu.pl/pub/GUST/contrib/BachoTeX98/ttf2pf/>. Nota: Estos ficheros son programas PostScript y deben descargarse manteniendo la tecla **Shift** cuando haga clic en el enlace. De lo contrario su navegador puede intentar arrancar ghostview para verlos.

Los archivos importantes para esta tarea son:

- GS_TTF.PS
- PF2AFM.PS
- ttf2pf.ps

Todo este lío de mayúsculas y minúsculas en los nombres es porque se tiene en cuenta las shells de DOS. ttf2pf.ps utiliza el resto como mayúsculas, por lo que cualquier cambio de nombre debe tener esto en cuenta. (En realidad, GS_TTF.PS y PFS2AFM.PS son parte de la distribución de Ghostscript, pero se pueden usar como herramientas independientes. FreeBSD no incluye esta última.) También puede ser que las instale (usted) en /usr/local/shared/groff_font/devps(?).

afmtodit

Crea archivos de tipos para usar con Groff desde el archivo de métricas de tipo ascii. Por lo general se encuentra en el directorio /usr/src/contrib/groff/afmtodit, pero hay unas cuantas cosas que hacer antes de poder usarlas.



Si cree que trabajar en /usr/src no es muy buena idea puede copiar el contenido del directorio anterior en otra ubicación.

En el directorio, necesitará compilar la utilidad. Escriba:

```
# make -f Makefile.sub afmtodit
```

Es posible que tenga que copiar también /usr/contrib/groff/devps/generate/textmap a /usr/shared/groff_font/devps/generate si no existe.

Una vez que todas estas utilidades estén en su sitio, estará listo para comenzar:

1. Cree el archivo .afm escribiendo:

```
% gs -dNODISPLAY -q -- ttf2pf.ps nombre_TTF nombre_tipo_PS nombre_AFM
```

Donde, *TTF_name* es su archivo de tipo TrueType, *PS_font_name* es el nombre del archivo .pfa, *AFM_name* es el nombre que quiere que tenga el archivo .afm. Si no especifica los nombres de los archivos de salida para los archivos .pfa o .afm, los nombres predeterminados se generan a partir del nombre de archivo de la tipo TrueType.

Esto también produce un archivo .pfa, el archivo ascii de las métricas del tipo PostScript (.pfb es para el formato binario). Esto no será necesario, pero podría (creo) ser útil para un servidor de tipos.

Por ejemplo, para convertir el tipo para código de barras 3of9 usando los nombres de archivo predeterminados use el siguiente comando:

```
% gs -dNODISPLAY -- ttf2pf.ps 3of9.ttf
Aladdin Ghostscript 5.10 (1997-11-23)
Copyright (C) 1997 Aladdin Enterprises, Menlo Park, CA. All rights reserved.
This software comes with NO WARRANTY: see the file PUBLIC for details.
Converting 3of9.ttf to 3of9.pfa and 3of9.afm.
```

Si desea que los tipos convertidos se almacenen en A.pfa y B.afm use este comando:

```
% gs -dNODISPLAY -- ttf2pf.ps 3of9.ttf A B
Aladdin Ghostscript 5.10 (1997-11-23)
Copyright (C) 1997 Aladdin Enterprises, Menlo Park, CA. All rights reserved.
This software comes with NO WARRANTY: see the file PUBLIC for details.
Converting 3of9.ttf to A.pfa and B.afm.
```

2. Crear el archivo PostScript Groff:

Vaya al directorio /usr/shared/groff_font/devps para que sea más fácil de ejecutar el siguiente comando. Probablemente necesitará privilegios de root. (O bien, si no se siente cómodo de todo trabajando en ese directorio, asegúrese de hacer referencia a los archivos DESC, text.enc y generate/textmap que están en el directorio).

```
% afmtodit -d DESC -e text.enc file.afm generate/textmap nombre_tipo_PS
```

Donde, file.afm es el *AFM_name* creado anteriormente por *ttf2pf.ps* y *PS_font_name* es el nombre del tipo utilizado para ese comando, así como el nombre que *groff(1)* utilizará para las referencias a este tipo. Por ejemplo, suponiendo que haya utilizado el comando *ttf2pf.ps* anterior, el tipo para código de barras 3of9 se puede crear usando el comando:

```
% afmtodit -d DESC -e text.enc 3of9.afm generate/textmap 3of9
```

Asegúrese de que el archivo *PS_font_name* resultante (por ejemplo, 3of9 en el ejemplo anterior) se encuentra en el directorio /usr/shared/groff_font/devps al copiarlo o moverlo allí.

Tenga en cuenta que si ttf2pf.ps asigna un nombre de tipo con el nombre que se encuentra en el archivo de tipos TrueType y quiere usar un nombre diferente antes de ejecutar **afmtodit** tiene que editar el archivo .afm. Este nombre también debe coincidir con el que se usa en el archivo Fontmap si desea redirigir [groff\(1\)](#) a [gs\(1\)](#).

9. ¿Se pueden usar los tipos TrueType con otros programas?

Windows, Windows 95 y Mac utilizan el formato de tipo TrueType. Es bastante popular y hay una gran cantidad de tipos disponibles en este formato.

Por desgracia conozco pocas aplicaciones que puedan usar este formato: me vienen a la mente Ghostscript y Povray. Según la documentación el soporte de Ghostscript es rudimentario y es probable que los resultados sean pobres comparados con los tipos Type 1. La versión 3 de Povray también tiene la capacidad de usar tipos TrueType, pero dudo que muchas personas creen documentos como una serie de páginas con trazado de rayos :-).

Esta situación, un tanto triste, puede cambiar pronto. El [proyecto FreeType](#) está desarrollando actualmente un conjunto útil de herramientas FreeType:

- El servidor de tipos para X11, **xfstt**, ofrece tanto tipos TrueType como tipos normales. Actualmente está en versión beta, pero dicen que es bastante útil. Consulte la [página de Juliusz Chroboczek](#) para más información. Las instrucciones para portarlo a FreeBSD se pueden encontrar en la [página de software de Stephen Montgomery](#).
- **xfstt** es otro servidor de tipos para X11 y está disponible en <http://sunsite.unc.edu/pub/Linux/X11/fonts/>.
- Un programa llamado **ttf2bdf** puede producir archivos BDF adecuados para su uso en un entorno X a partir de archivos TrueType. Los binarios para Linux están disponibles en <ftp://crl.nmsu.edu/CLR/multiling/General/>.
- y muchas más.

10. ¿Dónde se pueden obtener tipos adicionales?

Hay muchos tipos disponibles en Internet. Son totalmente gratuitos o shareware. Además, muchos de esos tipos están disponibles en la categoría de ports x11-fonts/

11. Preguntas adicionales

- ¿Para qué sirven los archivos .pfm?
- ¿Se puede generar el archivo .afm desde un archivo .pfa o .pfb?
- ¿Cómo generar los archivos de mapeo de caracteres Groff para tipos PostScript con nombres de caracteres no estándar?
- ¿Se pueden configurar los dispositivos xditview y devX para acceder a todos los tipos nuevos?
- Sería bueno tener ejemplos del uso de tipos TrueType con Povray y Ghostscript.