fuente_ISO-COMO

Urko Lusa Oiza, ulusa@frodo.com
v0.2, rev 2. 15 Oct 1996

Este documento explica cómo usar una fuente `iso-8859-1` en el modo texto de Linux, en vez de la que éste trae por defecto en todas las distribuciones que yo he visto, y que no deja de ser una especie de apaño que arrastra el lastre de la página de códigos de IBM.

Índice General

1 Introducción

Las distribuciones de Linux que conozco traen por defecto la misma fuente que el DOS (página de códigos 437). Para hacer que el sistema funcione como si la fuente fuera *ISO*, se emplea un extraño *"remapeo"* para hacer que al representarse determinado carácter, no salga por pantalla el que en la fuente ocupa dicha posición.

Para saber si este es el caso de tu sistema, mira el siguiente carácter:
Á
Si estás viendo una A mayúscula con acento agudo, este documento no te hace falta. Si ves un cuadrado, te puede interesar. Por supuesto lo anterior se aplica si estás mirando este documento en su versión texto desde la consola, o bien la versión `html` desde el `lynx`. Si lo estás viendo desde X o leyendo una versión impresa, deberías ver la A en cualquier caso.

También puedes probar `man iso_8859_1` desde la consola (según tu distribución y de si instalaste el manual o no, es posible que no tengas esta página) y comprobar si los caracteres mostrados se corresponden con su descripción y si ves correctamente **todos** ellos. Es muy posible que gran parte aparezcan como cuadrados en vez de corresponderse con la descripción que traen a la derecha.

Si quieres disponer de todos los caracteres necesarios para una correcta escritura de la lengua castellana (y varias otras), puedes seguir los pasos que se describen en este documento. Para los recelosos, comentar que esto no implica la pérdida de símbolos gráficos, y que si se hace bien, seguiremos viendo los cuadraditos en nuestras aplicaciones favoritas (`mc`, `slrn`, etc.).La única pega es que las líneas horizontales no salen unidas, cosa que creo que se debe a que la VGA sólo hace esto para ciertas posiciones de la tabla de caracteres. No obstante puedo asegurar que se puede sobrevivir sin ello.

A lo largo del documento aparecen varios ejemplos de comandos. Cuando éstos empiezan por `#` significa que quien debe ejecutarlos es el superusuario (`root`). Cuando aparecen precedidos por `$` significa que pueden ser ejecutados por cualquier usuario.

2 Preparando las VT

Lo primero que se debe hacer es lograr que cuando pulsemos ALT+`n`, salga por pantalla el carácter que ocupa el lugar `n` en la fuente y no otro. Esto se consigue con el siguiente comando:

```
$ echo -ne '\033(K'
```

pero por supuesto no vamos a estar tecleando eso en cada terminal virtual cada vez que tengamos que usarla, así que lo más cómodo es añadir lo siguiente en `/etc/rc.d/rc.local`:

```
# el BloqNum y los 8 bits
   for tty in /dev/tty[1-6]*; do
        setleds -D +num > $tty
        echo -n -e '\033(K' > $tty
   done
```

La línea que contiene `setleds` no es necesaria para el propósito de este documento, pero sería una pena no aprovechar el bucle ;-)

Lo siguiente que debemos hacer es instalar la fuente:

```
# setfont /usr/lib/kbd/consolefonts/lat1-16.psf
```

Para hacer que ésto ocurra automáticamente al arrancar, pon la orden en `/etc/rc.d/rc.font`. Puedes probar a cambiar a dicha fuente, teclear `echo -ne '\033(K'` y volver a la sección 1. Ahora deberías ver correctamente el carácter. Recuerda que el `echo` sólo tiene efecto sobre la terminal en la que se teclea, por lo que tendrás que teclearlo en la misma en la que estás leyendo esto.

3 Adaptando `terminfo`

La definición para la terminal de Linux que trae `ncurses` está pensada para usar los códigos de la `cp437`. Esto provocará que al cambiar de fuente veamos letras en vez de los símbolos gráficos. Afortunadamente esto no es difícil de solucionar cambiando la definición de terminal en `/usr/lib/terminfo/l/linux`.

NOTA: si tu distribución de Linux es antigua, es posible que la definición de terminal esté en `/usr/lib/terminfo/c/console`. Teclea

```
$ echo $TERM
```

para saber cuál te corresponde.

Primeramente deberemos extraer el fichero binario en forma de texto editable:

```
# infocmp > linux.original.tic
# cp linux.original.tic linux.tic
```

NOTA: si tu distribución de Linux es antigua, es posible que el equivalente a `infocmp` se llame `untic`.

NOTA2: asegúrate de estar en la consola normal (modo texto). Si estás en un xterm, será ésta la información que se vuelque al fichero.

Guarda el fichero `linux.original.tic` por si es necesario volver a restaurar la definición original, y edita los cambios sobre `linux.tic`. Éstos son los siguientes:

- Busca las siguientes entradas y bórralas:
  - `smacs`
  - `rmacs`
  - `acsc`

  IMPORTANTE: Las entradas están delimitadas por comas y puede haber varias en una misma línea. Ten cuidado en borrar **sólo** las entradas que se mencionan arriba.

- Añade las siguientes líneas:

  ```
  smacs=\E(K,
  acsc=l\206m\203k\214j\211u\215t\207v\213w\216q\212x\205,
  rmacs=\E)K,
  ```

- Graba los cambios y sal del editor.

Ya puedes proceder a compilar:

```
# tic linux.tic
```

Tras lo cual ya debería estar listo el sistema.

3.1 Probando si funciona

Seguramente tendrás el Midnight Commander. Ejecútalo para ver si los cambios han tenido efecto.

¡Pero si sale hecho un cristo!

Calma, calma. Pulsa `F9` y entra en el menú *Options*. Selecciona *Display bits* y elige la opción *Full 8 bits*. Asegúrate de pasar por el *Ok* o darle con el ratón. ¿Mejor ahora? Vuelve a entrar en el menú y graba los cambios con *Save setup*.

En caso de que estés leyendo esto con `lynx`, debes saber que en las fuentes de este navegador viene una página que te será muy útil para ver los cambios que has realizado. Dicha página se llama `iso8859-1.html`, y si estás leyendo esto conectado puedes verla en `http://www.uni-passau.de/~ramsch/iso8859-1.html`

3.2 Nota sobre `slang`

Las versiones de `slang` anteriores a la `0.99-32` no determinan cuáles son los caracteres gráficos leyendo `terminfo`, sino que usan unos predeterminados (si la terminal es linux). Si tu versión de esta librería es anterior, puedes actualizarte cogiendo la última versión de `ftp://space.mit.edu/pub/davis/slang`.

Algunos programas que usan `slang` son `slrn`, `jed` y `lynx` (si se compila con esta opción).

3.3 Nota sobre `reset`

Algo que no sé cómo evitar es que al ejecutar `reset` los carateres vuelvan a ser "remapeados" como si la fuente fuese la `cp437`. El único apaño que se me ocurre es incluir esta línea en `/etc/profile` o en el `.profile` del usuario:

```
alias reset='reset ; echo -ne "\033(K"'
```

4 Más gaitas

Mientras esperamos el día en que las distribuciones de Linux traigan las diferentes definiciones de `locale` para su internacionalización, hay algunas cosillas que podemos afinar.

4.1 Variables de entorno

Puedes ponerlas en `/etc/profile` o en tu `.profile` particular. He seguido la sintaxis de `bash`. Si usas otro shell posiblemente tengas que cambiar `export` por la orden equivalente.

4.1.1 `less`

Para visualizar con `less` ficheros que contengan caracteres de 8 bits:

```
export LESSCHARDEF=8bcccbcc18b.
```

Esto hará que se consideren como imprimibles **todos** los caracteres desde el 32 hasta el 255, lo que puede acarrear problemas si se usa en una terminal `vt200`, por ejemplo, pero permite ver caracteres gráficos desde el PC. Si tienes una terminal en `ttySn` puedes poner esto:

```
export LESSCHARDEF=8bcccbcc18b.
if [ "`tty`" = "/dev/ttySn" ]
```

```
then
   TERM=vt220d
   unset LESSCHARDEF
   export LESSCHARSET=latin1
else
   TERM=linux
fi
export TERM
```

lo que hará que si se entra por `ttySn`, `less` no considere como imprimibles los caracteres del 128 al 160 (los caracteres gráficos en la fuente `lat1-16.psf`, pero que son interpretados como caracteres de control por el `vt200`).

4.1.2 `LANG`, `LC_CTYPE`, etc...

Estas variables son las que se usan para presentar el formato de la fecha, orden alfabético, etc. Desgraciadamente que yo sepa aún no están implementadas en las distribuciones habituales de Linux. De todas formas puedes añadir esto:

```
export LANG=es_ES
```

4.2 `isprint`

Esta es la función del sistema que decide si un carácter es imprimible o no. Si la distribución soportase `locale`, debería funcionar bien, pero tal como está ahora la cuestión, sólo devuelve como imprimibles los caracteres `ASCII`.

Esto provoca que muchos programas compilados con `#include <ctype.h>` no muestren correctamente los caracteres no ASCII. Como esto puede cambiar en un futuro, es importante que antes de aplicar el cambio que se propondrá a continuación compruebes tu sistema con el siguiente programa:

```
#include <locale.h>
#include <ctype.h>

void main(void)
{
unsigned char c;

setlocale(LC_CTYPE,"");

for (c=191;c<211;c++)
        {
        printf("%d - %c ",c,c);
        if (isprint(c)) puts("es imprimible");
        else puts("no es imprimible");
        }
}
```

Graba el texto anterior con el original nombre de `prueba.c` y compílalo con

```
$ gcc -s -o prueba prueba.c
```

y pruébalo así:

```
$ export LC_CTYPE=ISO-8859-1
$ ./prueba
```

Seguramente saldrán todos los caracteres como no imprimibles. Por si acaso asegúrate antes probando con otros valores de `LC_CTYPE`, tales como `es_ES`, `fr_FR`, `pt_PT`, etc. Para saber más posibles valores de `LC_CTYPE`, consulta el fichero `locale.alias`, que tendrás (si instalaste X) por `/var/X11R6/lib/locale` o algún otro directorio similar.

Si, y **sólo si**, el programa `prueba` clasifica como no imprimibles los caracteres que saca, puedes hacer el siguiente cambio en `/usr/include/ctype.h`.

Primero haz copia de seguridad del fichero:

```
# cd /usr/include
# cp -a ctype.h ctype.h.orig
```

En la línea 150 aprox. cambia

```
#define      isprint(c)      __isctype((c), _ISprint)
```

por

```
#define isprint(c)      (((c & 0xff) == 0x1b) || ((c & 0xff) > 0x1f))
```

Este cambio de `ctype.h` fue publicado por Pablo Saratxaga <`srtxg@f2219.n293.z2.fidonet.org`> en el difunto grupo `es.foro.unix`. Yo he recompilado con él algunos programas como `ncftp` y `tin`, y ha funcionado muy bien.

4.3 Configuración de algunos programas

4.3.1 dosemu

Si arrancamos dosemu en modo texto, los caracteres que ocupen una posición mayor que el 127 ya no se verán bien. Para remediarlo se puede habilitar el acceso a la VGA, con lo que `dosemu` usará el juego de caracteres de nuestra tarjeta gráfica, que es del tipo IBM, por lo menos si usamos Linux en un ix86. Esto se consigue poniendo lo siguiente en el fichero de configuración (generalmente `/etc/dosemu.conf`):

```
video { vga  console  graphics }
```

También es conveniente que tengamos las siguiente opción:

```
keyboard {  layout es-latin1  keybint on  rawkeyboard on  }
```

Con esto y el programa `keybes.com` de Pedro Maicas, no debería haber ningún problema para ver y teclear los caracteres mayores de 127.

4.3.2 pine

Se puede cambiar la configuración desde el propio programa, o bien editar el siguiente cambio en el fichero `~/.pinerc`:

```
character-set=ISO-8859-1
```

Desde la versión 3.92, Pine ya permite poner artículos con 8 bits en los grupos de noticias (en el correo dependerá de la versión de sendmail y de cómo esté configurado). Para habilitar esta opción, añadir `enable-8bit-nntp-posting` a la lista `feature-list` del fichero mencionado.

Existe también un fichero que efectúa cambios globales (para todos los usuarios del sistema). El fichero se llama `pine.conf`, y su ubicación depende de la distribución que usemos.

4.3.3 ELM

Añadir lo siguiente a `~/.elm/elmrc`:

```
charset=ISO-8859-1
displaycharset=ISO-8859-1
textencoding = 8bit
```

También se pueden hacer estos cambios para todos los usuarios. Basta añadir lo mismo al fichero general de configuración, que en la Slackware está en `/var/lib/elm/elm.rc`.

4.3.4 TIN

Añadir lo siguiente a `~/.tin/headers`:

```
Mime-Version: 1.0
Content-Type: text/plain; charset=iso-8859-1
Content-Transfer-Encoding: 8bit
```

Algunas versiones modernas, como la 1.3 unoff, ya incorporan este tipo de cabeceras, por lo que la edición de este fichero es innecesaria.

5 Problemas

A ver qué **has** hecho mal ahora... `};-)`

5.1 La fuente no se carga al arrancar

Asegúrate de que el script `/etc/rc.d/rc.font` es invocado desde `/etc/rc.d/rc.M`. Si `rc.font` no aparece mencionado en `rc.M`, añade a este último lo siguiente:

```
if [ -r /etc/rc.d/rc.font ]; then
  . /etc/rc.d/rc.font
  fi
```

`rc.font` debería tener permiso de ejecución (`chmod +x rc.font`).

5.2 El teclado funciona mal en VT(no.)

Asegúrate de que el bucle de `/etc/rc.d/local` incluye todas las terminales virtuales que vayas a usar. Es decir, si tienes 8, sustituye

```
for tty in /dev/tty[1-6]*;
```

por

```
for tty in /dev/tty[1-8]*;
```

5.3 Los caracteres gráficos no aparecen correctamente

El cambio propuesto para `terminfo` está pensado para la fuente lat1-*xx*.psf (donde *xx* representa la resolución: 8, 10...). Otras fuentes, como `iso01.f16`, no tienen caracteres gráficos (compruébalo ejecutando `showfont`). Y otras pueden tenerlos en diferentes posiciones (como las iso01a).Si estás en el segundo caso, puedes cambiar la entrada `acsc` de `terminfo` incluyendo los valores en octal para los diferentes símbolos gráficos. Consulta `man terminfo` para conocer la sintaxis.

5.4 ¿Cómo se quita esto?

No puedo creer que me hagas esto. Pero en fin, si has seguido los pasos correctamente, no deberías tener problemas en dar marcha atrás. Haz lo siguiente desde el directorio donde tengas `linux.original.tic`.

```
# tic linux.original.tic
# rm /usr/include/ctype.h
# mv /usr/include/ctype.h.orig /usr/include/ctype.h
# rm /etc/rc.d/rc.font
```

6 Sobre este documento

Este documento ha sido elaborado utilizando `Linuxdoc-SGML v1.5.`

Si lo has seguido fielmente habrás conseguido una de estas dos cosas:

1. Adaptar la fuente de tu sistema al estándar ISO (más o menos) en vez de usar un sistema propietario, recuperando unos cuantos símbolos muy convenientes o...
2. Descuajeringar totalmente tu configuración.

Si estás en el primer caso y tienes alguna idea más a este respecto que te gustaría compartir, no dudes en escribirme a `ulusa@frodo.com` (Fidonet: 2:343/143.30)

Si estás en el segundo, no te molestes en escribirme. Mi ego no lo soportaría.

En caso de que estés leyendo este documento localmente y quieras ver la última versión, es posible que esté en

`http://www.arrakis.es/~ulusa/linux/fuente_ISO-COMO.` *extensión*

donde *extensión* puede ser:

- `html`: versión html.
- `sgml.gz`: fuente sgml comprimida.
- `dvi.gz`: fichero dvi comprimido.
- `ps.gz`: fichero PostScript comprimido.

También puedes obtener la fuente que uso en

`http://www.arrakis.es/~ulusa/linux/swiss-3-lat1.fnt.gz`

Es muy posible que lo que se trata aquí ya esté explicado en algún *HOWTO* de los que trae el Linux. No lo sé con seguridad porque nunca los leo ;-)

6.1 Agradecimientos

A Francisco José Montilla, `pacopepe@insflug.org`.

A Pablo Saratxaga, `srtxg@f2219.n293.z2.fidonet.org` por su *hack* de `isprint`.

A todos aquellos que me han enviado comentarios.

7 Anexo: El INSFLUG

El *INSFLUG* forma parte del grupo internacional *Linux Documentation Project*, encargándose de las traducciones al castellano de los Howtos (Comos), así como la producción de documentos originales en aquellos casos en los que no existe análogo en inglés.

En el **INSFLUG** se orienta preferentemente a la traducción de documentos breves, como los *COMOs* y *PUFs* (**P**reguntas de **U**so **F**recuente, las *FAQs*. `:)`), etc.

Diríjase a la sede del INSFLUG para más información al respecto.

En la sede del INSFLUG encontrará siempre las **últimas** versiones de las traducciones: `www.insflug.org`. Asegúrese de comprobar cuál es la última versión disponible en el Insflug antes de bajar un documento de un servidor réplica.

Se proporciona también una lista de los servidores réplica (*mirror*) del Insflug más cercanos a Vd., e información relativa a otros recursos en castellano.

Francisco José Montilla, `pacopepe@insflug.org`.