Vesafb Mini-COMO

Autor: Alex Buell, *alex.buell@tahallah.demon.co.uk* Traducido por: Jesús Martínez Mateo, *jmartinez@sportec.es*
v0.5, 2 de Agosto de 1998

Este documento describe cómo usar el dispositivo vesafb bajo Linux con una tarjeta gráfica compatible VESA 2.0 en plataformas Intel.

Índice General

1 Aportaciones

Gracias a todas las personas nombradas a continuación por su ayuda en la mejora de este documento (*Vesafb Mini-HOWTO*).

- Jeff Noxon *jeff@planetfall.com*
- Francis Devereux *f.devereux@cs.ucl.ac.uk*
- Andreas Ehliar *ehliar@futurniture.se*
- Martin McCarthy *marty@ehabitat.demon.co.uk*
- Simon Kenyon *simon@koala.ie*
- David Ford *david@kalifornia.com*
- Chris Negro *cblack@cmpteam4.unil.ch*

- N Becker *nbecker@fred.net*
- Bob Tracy *rct@gherkin.sa.wlk.com*
- Marius Hjelle *marius.hjelle@roman.uib.no*
- Aaron Tiensivu *tiensivu@pilot.msu.edu*

y por supuesto los autores de los dispositivos framebuffer:

- Gerd Knorr *kraxel@cs.tu-berlin.de*
- Geert Uytterhoeven *Geert.Uytterhoeven@cs.kuleuven.ac.be*
- Martin Mares *mj@ucw.cz*

Y a todas aquellas personas que deberían aparecer en este documento y no aparecen. :o)

2 ¿Qué es vesafb?

Vesafb es un controlador framebuffer para la arquitectura Intel que trabaja con las tarjetas gráficas compatibles con la norma VESA 2.0. Este *driver* se encuentra estrechamente relacionado dentro del núcleo con los controladores de dispositivo para framebuffer.

Vesafb es un *driver* de pantalla que habilita el uso de modos gráficos en la plataforma Intel para las consolas de texto mediante mapas de bits (*bitmapping*). También permite desplegar un logotipo que probablemente es la razón principal por la que quiere usar vesafb `:o)`

Desgraciadamente, no puede usar vesafb con las tarjetas que soportan la norma VESA 1.2. Esto es debido a que estas tarjetas no usan *framebuffering*. Puede ser que alguien decida escribir un driver de dispositivo (`vesafb12`) para estas tarjetas, pero esto usaría la preciada memoria del núcleo. `:o(`

Sin embargo, existe la posibilidad de habilitar extensiones VESA 2.0 para tarjetas VESA 1.2.

Puede conseguir un tipo de programa TSR que se ejecutará desde DOS, y usado en conjunción con loadlin, puede ayudar a configurar la tarjeta para los modos gráficos de consola apropiados.

Tenga en cuenta que esto no siempre funcionará, por ejemplo algunas tarjetas como la serie VLB 54xx de Cirrus Logic son mapeadas en un rango de direcciones de memoria (por ejemplo, dentro del rango 15MB-16MB) que exigen al sistema tener más de 32MB de memoria para funcionar.

(Hay una manera de solucionar esto, si tiene una opción en la BIOS para dejar un hueco de memoria en el rango 15MB-16MB, podría funcionar, pero me han dicho que Linux no soporta esto). Si desea experimentar con esta opción, hay un completo repertorio de programas TSR disponibles (un ejemplo de ellos es UNIVBE) que pueden encontrarse en Internet.

3 ¿Qué es un dispositivo framebuffer?

Un dispositivo framebuffer es una abstracción para el hardware gráfico. Representa un buffer de imagen para el hardware de vídeo, y permite a las aplicaciones software acceder al hardware gráfico a través de una interfaz bien definida, de manera que el software no necesita saber nada sobre la interfaz de bajo nivel (Tomado de `/Documentation/fb/framebuffer.txt` de Geert Uytterhoeven en las fuentes del kernel de linux).

4 ¿Cómo activo los drivers vesafb?

(nota: La información esta basada en lo que sé y he logrado usando el núcleo de Linux `2.1.112` en un P75+ con una tarjeta gráfica ATI VideoExpress 2MB. No dude en enviarme un email con información o casos especiales para tarjetas gráficas diferentes).

Asumiendo que usa `menuconfig` necesitará realizar los siguientes pasos:

Entre en la sección *Code Maturity Level* del menú, y habilite la opción para desarrollo y/o drivers incompletos (*Development and/or Incomplete Drivers*) (nota: esto puede cambiar para núcleos futuros; cuando esto suceda, este HOWTO será revisado).

Entre en el submenú de dispositivos de consola (*Console Drivers*), y habilite lo siguiente:

-
- *VGA Text Console*
- *Video Selection Support*
- *Support for frame buffer devices (experimental)*
- *VESA VGA Graphic console*
- *Advanced Low Level Drivers*
- *Select Mono, 2bpp, 4bpp, 8bpp, 16bpp, 24bpp y 32bpp packed*
- *Pixel drivers*
- *VGA character/attributes support*

VGA Chipset Support (text only) (`vgafb`) era parte de la lista anterior, pero ha sido eliminado ya que está obsoleto y no será soportado. Se eliminará en breve. Use *VGA Text Console* en su lugar.

Asegúrese de que *Mac variable bpp packed pixel support* no está habilitado (A partir del `2.1.111`, esto parece habilitarse si los *Advanced Low Level Drivers* son seleccionados inicialmente). Asegúrese de que no sean módulos (No es seguro que sea posible compilarlos todavía como módulos; por favor corríjame si me equivoco en esto).

A continuación, recompile el núcleo, modifique el fichero `/etc/lilo.conf` de modo que incluya el parámetro `VGA=ASK`, y ejecute (instalando) `lilo`, esto permitirá seleccionar el modo deseado en el momento de arrancar.

Reinicie, y como prueba, introduzca la entrada `0301` tras la solicitud VGA (esto le dará una resolución de 640x480 a 256 colores), y deberá verse un logotipo con un pequeño Pingüino durante el arranque.

Una vez que pueda ver esto funcionando, puede explorar los distintos modos VESA (ver abajo) y decidir cuál es el que más le gusta, fije el parámetro "`VGA=x`" en el fichero `/etc/lilo.conf`. Cuando haya escogido el que más le guste, consulte el valor decimal equivalente de las tablas siguientes y use el número decimal correspondiente (es decir, para una resolución de 1280x1024 a 256 colores, debe usar el valor "`VGA=775`"), y reejecutar `lilo`. Esto es todo!.

Para referencias más extensas, leer los documentos relacionados con LoadLin y LILO.

5 ¿Qué modos VESA están disponibles?

Esto realmente depende del tipo de tarjeta gráfica compatible con la norma VESA 2.0 que tenga en su sistema, y de la cantidad de memoria de vídeo disponible. El procedimiento anterior es sólo una manera de comprobar qué modos funcionan mejor con su tarjeta gráfica.

La siguiente tabla muestra los números de los modos que puede utilizar en la solicitud del parámetro VGA (realmente, a estos modos se le han sumado el valor 0x200 para facilitar su referencia en la tabla):

Colores	640x480	800x600	1024x768	1280x1024	1600x1200
256	0301	0303	0305	0307	031C
32,768	0310	0313	0316	0319	031D
65,536	0311	0314	0317	031A	031E
16.8M	0312	0315	0318	031B	031F

La misma tabla expresada en términos decimales:

Colores	640x480	800x600	1024x768	1280x1024	1600x1200
256	769	771	773	775	796
32,768	784	787	790	793	797
65,536	785	788	791	794	798
16.8M	786	789	792	795	799

(El autor estaría agradecido si le pudiera proporcionar los identificadores `03xx` para los modos adicionales, esto es, 1152x900)

6 ¿Hay algún driver X11 para vesafb?

Sí, lo hay. Necesitará usar el driver `XF86_FBdev` si por alguna razón su driver X11 actual no es compatible con `vesafb`. Vaya a la página *http://www.xfree86.org*, y baje el archivo `X332servonly.tgz`, desempaquételo, configúrelo revisando `xc/config/cf/xf86site.def`, y active el `#define` para `XF68FBDevServer`. No se preocupe por la referencia m68k, esta soporta plataformas Intel. Recompílelo todo; tardará un tiempo considerable en función del tamaño del código fuente.

Todavía hay algo más, no hay disponible ningún `XF86_FBdev` binario precompilado, pero creo que Debian puede estar trabajando en esto; y ellos podrán producir las dos variantes, *libc5* y `glibc2`.

Ha habido informes de que X11 no funciona con ciertas tarjetas gráficas si esta característica vesafb está habilitada, si esto le ocurre, pruebe el nuevo driver `XF86_FBdev` para X11.

Este driver, junto con el `vesafb` también puede ayudar a ejecutar X11 en resoluciones gráficas mayores con algunos chipsets gráficos que no son soportados por los drivers X11 actuales. Algunos ejemplos son MGA G200 y similares.

Afortunadamente se solucionarán los problemas de X11 con el soporte de estas tarjetas gráficas en futuras versiones.

7 ¿Puedo compilar vesafb como módulo?

A partir del núcleo `2.1.112`, los vesafb no pueden ser modularizados, aunque con el tiempo, algún desarrollador de `vesafb` puede decidir modificar los fuentes para permitir la modularización. Tenga en cuenta que incluso si fuera posible la modularización, durante el arranque no podría ver nada en la pantalla hasta que el driver `vesafb` fuera cargado (vía `modprobe` o mediante una imagen `initrd`). Por esta razón, es una decisión inteligente dejarlo en el núcleo, para aquellos casos en los que existan problemas de arranque.

8 Miscelánea y Advertencias

Esto sólo es válido para versiones del núcleo 2.1.112.Este COMO será continuamente actualizado siguiendo los progresos de desarrollo del núcleo. No dude en mandarme un email con información o críticas :o)

Otro asunto es que el *scrollback buffering* no funciona todavía. Puede que sea solucionado en futuras versiones.

9 Acerca de la traducción

Este documento ha sido traducido por el grupo *Linux-Kfe*, integrado en el INSFLUG. Puede visitar nuestra página en la siguiente dirección: *http://members.es.tripod.de/linuxkfe* (también puede utilizar el enlace: *http://go.to/linuxkfe*).

Actualmente gestionamos este grupo las siguientes personas:

Director del Proyecto:

Jesús Martínez Mateo *jmartinez@sportec.es*, *kfetro@cajamurcia.es*

Coordinadores:

- Juan José Roldán Guerrero *jroldan@sportec.es*
- Pedro Ruiz Maldonado *bigplac@newmail.net*

Si desea colaborar con nosotros en futuras traducciones y distintos proyectos relacionados con el mundo de Linux (o si simplemente desea realizar alguna aportación, crítica o sugerencia), envíe un email a la siguiente dirección: *linuxkfe@lycosmail.com*. Procuraremos atenderle lo antes posible, de todas formas, si lo cree conveniente puede dirigirse de forma directa a cualquiera de los coordinadores nombrados anteriormente.

Fecha de la última modificación: 28 de Noviembre de 1999.

10 Anexo: El INSFLUG

El *INSFLUG* forma parte del grupo internacional *Linux Documentation Project*, encargándose de las traducciones al castellano de los Howtos, así como de la producción de documentos originales en aquellos casos en los que no existe análogo en inglés, centrándose, preferentemente, en 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 ella encontrará siempre las **últimas** versiones de las traducciones «oficiales»: *www.insflug.org*. Asegúrese de comprobar cuál es la última versión disponible en el Insflug antes de bajar un documento de un servidor réplica.

Además, cuenta con un sistema interactivo de gestión de fe de erratas y sugerencias en línea, motor de búsqueda específico, y más servicios en los que estamos trabajando incesantemente.

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

En *http://www.insflug.org/insflug/creditos.php3* cuenta con una detallada relación de las personas que hacen posible tanto esto como las traducciones.

¡Diríjase a *http://www.insflug.org/colaboracion/index.php3* si desea unirse a nosotros!.

«Cartel» Insflug, *cartel@insflug.org*.