

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
 Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
 HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
 Gestor de contenidos (Alfresco)
 Aplicaciones híbridas

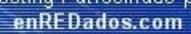
Tareas programadas (Quartz)
 Gestor documental (Alfresco)
 Inversión de control (Spring)

Control de autenticación y
 acceso (Spring Security)
 UDDI
 Web Services
 Rest Services
 Social SSO
 SSO (Cas)

JPA-Hibernate, MyBatis
 Motor de búsqueda empresarial (Solr)
 ETL (Talend)

Dirección de Proyectos Informáticos.
 Metodologías ágiles
 Patrones de diseño
 TDD

BPM (jBPM o Bonita)
 Generación de informes (JasperReport)
 ESB (Open ESB)

| | |
|---|---|
|  Powered by  | Hosting Patrocinado por   |
|---|---|

[Home](#) | [Quienes Somos](#) | [Empleo](#) | [Tutoriales](#) | [Contacte](#)



CoNcept

Lanzado

TNTConcept versión 0.4.1 (04/06/2007)

Desde [Autentia](#) ponemos a vuestra disposición el software que hemos construido (100% gratuito y sin restricciones funcionales) para nuestra gestión interna, llamado TNTConcept (auTeNTia).

Construida con las últimas tecnologías de desarrollo Java/J2EE (Spring, JSF, Acegi, Hibernate, Maven, Subversion, etc.) y disponible en licencia GPL, seguro que a muchos profesionales independientes y PYMES os ayudará a organizar mejor vuestra operativa.

Las cosas grandes empiezan siendo algo pequeño Saber más en: <http://tntconcept.sourceforge.net/>

| | |
|--|---|
| <p style="text-align: center;">Tutorial desarrollado por: Alejandro Perez García 2003-2007 Alejandro es Socio fundador de Autentia y nuestro experto en J2EE, Linux y optimización de aplicaciones empresariales.</p> <p>Si te gusta lo que ves, puedes contratarle para impartir cursos presenciales en tu empresa o para ayudarte en proyectos (Madrid).</p> <p style="text-align: center;">Contacta: alejandropg@autentia.com.</p> | <p>NUEVO CATÁLOGO DE SERVICIOS DE AUTENTIA (PDF 6,2MB)</p> <p>www.adictosaltrabajo.com es el Web de difusión de conocimiento de www.autentia.com</p>  <p>autentia real business solutions</p> <p style="text-align: center;">Catálogo de cursos</p> |
|--|---|

Descargar este documento en formato PDF [debianInstall.pdf](#)

[Firma en nuestro libro de Visitas <----->](#) [Asociarme al grupo AdictosAlTrabajo en eConozco](#)

Nuevas Herramientas Web

¿Quieres que los usuarios de tu sitio recomienden tuweb TE AYUDAMOS
www.santiago-corp.org

Virtualización Xen

Traslada las ventajas del software libre a su infraestructura de IT

Fecha de creación del tutorial: 2007-06-05

Instalar Debian en un Core 2 Duo (Asus G1)

Índice de contenidos

- [1. Introducción](#)
- [2. Entorno](#)
- [3. Descarga e Instalación](#)
- [4. Actualización del sistema](#)
- [5. Compilación del kernel](#)
 - [5.1. Instalación de paquetes para compilar el kernel](#)
 - [5.2. Configurando yaird](#)
 - [5.2. Configuración del kernel](#)
 - [5.3. Compilación del kernel](#)
 - [5.4. Instalación del kernel](#)
- [6. Instalación del driver para la nVidia](#)
- [7. Instalación del driver para la wireless](#)
- [8. Conclusiones](#)
- [9. Sobre el autor](#)

1. Introducción

Hace poco he cambiado de portátil y como siempre he ido a instalar como sistema operativo Debian GNU / Linux (www.debian.org).

Hasta ahora uno de los puntos débiles de Debian había sido la instalación ya que en muchas ocasiones no se conseguía terminar con éxito a no ser que se fuera un experto, y esto echaba a mucha gente para atrás. En esta ocasión parece que los chicos de Debian se están superando ya que no podía ser más fácil.

En este tutorial vamos a hacer un repaso a la instalación.

2. Entorno

El tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil Asus G1 (Core 2 Duo a 2.1 GHz, 2048 MB RAM, 120 GB HD).
- Sistema Operativo: GNU / Linux, Debian Sid (unstable), Kernel 2.6.20, KDE 3.5

3. Descarga e Instalación

Para instalar Debian tenemos básicamente dos opciones:

1. Usar la instalación completa: podemos bajarnos todos los CDs (pueden ser unos 23 CDs) o un DVD.
2. Bajarnos una instalación mínima e instalar el resto por Internet: Dentro de esta opción la más típica es la instalación "netinst" donde nos bajamos una imagen de entre 135 y 175 MB, pero tenemos otras opciones como por ejemplo usb stick.

En mi caso siempre instalo con la opción "netinst" de la opción 2, ya que sólo te bajas de internet lo que realmente te interesa.

En esta ocasión he elegido la versión diaria de la "testing" que te puedes descargar desde: <http://www.debian.org/devel/debian-installer/>. En esta página podemos encontrar versiones "snapshot" que van actualizando con bastante frecuencia. Esta versión ya está bastante conseguida (instalación gráfica y detección de más hardware), así que merece la pena usarlas.

Yo elegí la instalación para i386 (creo que todavía no está suficientemente maduro el tema de los 64 bits, ya que todavía hay cosas que no funcionan correctamente, así que todavía no me atrevo ;). En concreto me bajé:

```
Debian 'etch' installation CD-ROM
Built: 20070406-21_06; d-i 20070308
```

Una vez te bajas la imagen hay que grabarla en un CD. Una vez tenemos el CD listo, basta con meter el CD en la unidad del ordenador donde queremos instalar, y arrancar la máquina (por supuesto en la BIOS debe estar habilitado el arranque del ordenador desde CD).

Nada más arrancar nos aparece en la pantalla el logo de Debian y un breve mensaje. Podemos dar a "ENTER" para arrancar directamente o F1 para ver la ayuda.

Yo elegí F1, y luego F3 para ver los posibles métodos de instalación:

- install: la instalación de toda la vida
- installgui: la instalación gráfica.
- expert: la instalación para los expertos.
- expertgui: la instalación gráfica para los expertos.

Las opciones para expertos no os las recomiendo a no ser que realmente seáis unos verdaderos expertos porque os va a hacer muuuuuchas preguntas. Así que mejor escogemos la instalación normal, y por supuesto, la versión gráfica del instalador.

```
installgui tasks="kde-desktop, standard"
```

Las opciones que hemos dado en el parámetro "tasks" es para instalar KDE, ya que por defecto se instala GNOME, y yo prefiero KDE (aunque sobre gustos no hay nada escrito ;).

Le damos a "ENTER" y la instalación empieza de forma inmediata:

- Nos pregunta el idioma para el proceso de instalación
- Nos pregunta la localización
- Nos pregunta la distribución de teclado
- Empieza a detectar cosas y a instalar cosas
- Detecta la red
- Nos pregunta el nombre de la máquina
- Nos pregunta el dominio
- Detecta los discos
- Hacer las particiones: Tenemos varias opciones guiadas y una opción manual. Para servidores es muy recomendable hacer distintas particiones para /boot, /bin, /home, /var, ... y de hecho hay varias teorías y documentación de como se deben crear las particiones. Pero para un equipo de uso personal esto no tiene tanta importancia. Yo elegí la opción guiada que coge el tamaño libre más grande (para respetar la partición de windows que en mi trabajo en Autentia www.autentia.com a veces me resulta muy útil) y usé una sola partición para montar "/" , y otra partición para "swap" (deje todas las opciones que me propuso el instalador).

- Nos pregunta la zona horaria
- Nos pregunta la clave de root
- Nos crea una cuenta de usuario
 - Nombre completo
 - Nombre de la cuenta
 - Clave
- Instala el sistema base
- Seguimos instalando con un mirror de Internet. Muy recomendable porque de lo contrario nos quedaremos con un sistema mínimo, la idea es, a partir de ahora, instalar los paquetes descargándolos de Internet.
- Configuramos el proxy si fuera necesario.
- Nos pregunta si queremos participar en el concurso de popularidad. Si decimos que si, nuestro sistema enviará al equipo de Debian, información sobre los paquetes más utilizados. Esta información les ayuda a determinar que paquetes deberían ir en el primer CD de la distribución. Si decimos que no, no pasa nada.
- Ahora es cuando nos vamos a tomar un café porque va a tardar un ratito en descargar e instalar (con mi conexión de 4 megas, unos 20 minutos).
- Nos pregunta si queremos instalar el GRUB (aplicación que se encarga de gestionar el arranque del sistema) en el MBR (Master Boot Record). Aquí mi recomendación es NO, sobre todo si, como yo, tenéis un arranque dual con Windos como vecino (cuando se instala o reinstala tiende a machacar el MBR).
- Nos pregunta cual es la partición de arranque. Si no nos acordamos del nombre de la partición de arranque de Debian (lo hicimos la definir las particiones), con Ctrl + Alt + F2 podemos activar otra consola, con fdisk /dev/sda y luego opción "p" vemos todas las particiones. Luego para volver a la consola de la instalación Ctrl + Alt + F5
- Sacamos el CD y reiniciamos.

Con estos sencillos pasos, y en pocos minutos, ya tenemos instalado nuestro Debian con Kernel 2.6.18-4. Ha sido rápido e indoloro ;)

Al arrancar el sistema vemos que nos funciona la tarjeta gráfica nVidia, el sonido, la red, el touch pad, los usb, la grabadora de DVD.

A continuación vamos a ver como mejorar el soporte de nuestro hardware. Después de instalar el sistema vemos que la wireless no está funcionando, esto es algo importante que vamos a arreglar en seguida.

Algo de menor importancia es el tema de la tarjeta gráfica y el procesador. Aunque están funcionando correctamente, no estamos sacando todo el partido posible, así que, al final de este tutorial, acabaremos usando el driver proporcionado por nVidia y vamos a compilar el kernel para soportar nuestro procesador Intel Core 2 Duo,

4. Actualización del sistema

En Debian existen tres versiones principales "stable", "testing" y "unstable". Sus nombres indican claramente la fiabilidad de los paquetes que contienen. Aunque tengo que decir a favor de la "unstable" que llevo trabajando con ella varios años y rara vez he tenido algún problema: la mayoría de las veces son tonterías que a los dos o tres días arreglan al sacar una nueva actualización; y si se presenta algún problema realmente serio (creo que sólo me ha pasado un par de veces) basta con desactualizar los paquetes que lo causan y quedarnos como estábamos antes de la actualización.

Lo que realmente nos interesa de la "unstable" es que trae el kernel 2.6.20 y superiores, y en estas versiones del kernel es donde vamos a encontrar el soporte para el procesador Core 2 Duo.

Vamos al grano. Como root editamos el fichero **/etc/apt/sources.list** y ponemos el siguiente contenido:

```
#
# deb cdrom:[Debian GNU/Linux testing _Etch_ - Official Snapshot i386 NETINST Binary-1 20070406-21:06]/
# etch contrib main

# deb http://ftp.es.debian.org/debian/ lenny main
# deb-src http://ftp.es.debian.org/debian/ lenny main

# deb http://security.debian.org/ lenny/updates main contrib
# deb-src http://security.debian.org/ lenny/updates main contrib

deb http://ftp.debian.org/debian/ stable main non-free contrib
deb http://ftp.debian.org/debian/ testing main non-free contrib
deb http://ftp.debian.org/debian/ unstable main non-free contrib
```

Como veis hemos comentado los repositorios de paquetes del CD y de la versión lenny que es la que hemos instalado, y hemos añadido los repositorios de la "stable", "testing" y "unstable". El sistema de paquetes, de forma automática buscara la versión más nueva por todos los repositorios que tengamos dados de alta.

Para refrescar la información de los paquetes ejecutamos como root:

```
# apt-get update
```

Cuando termine, ejecutamos:

```
# apt-get -u dist-upgrade
```

Esto actualizará todos los paquetes de nuestro sistema a sus versiones más recientes. Va tardar un buen rato, ya que prácticamente va a actualizar el sistema entero, así que paciencia.

5. Compilación del kernel

En este punto vamos a ver como conseguir los paquetes necesarios y como compilar el kernel de nuestro sistema operativos.

5.1. Instalación de paquetes para compilar el kernel

Para instalar y compilar el kernel no es necesario hacer todas las tareas como `root`. Se pueden instalar paquetes como el `fakeroot` para hacer ciertas de estas tareas con un usuario cualquiera. Aunque mucha gente recomienda esto, yo voy a hacer todas las tareas como `root` para no complicar más el tema (eso si, tener cuidado de no borrar vuestra raíz de directorios ;)

Necesitaremos instalar (con `apt-get`, `synaptic` o cualquier otra herramienta de gestión de paquetes) los siguientes paquetes:

- **gcc**: el compilador
- **libncurses5-dev**: ficheros necesario para poder usar el menú de configuración del kernel
- **linux-tree-2.6.21**: el código fuente del kernel.
- **kernel-package**: scripts para facilitarnos enormemente al tarea de recompilar el kernel.
- **yaird**: para crear imágenes de arranque del sistema.
- **module-assistant**: para luego poder compilar drivers al margen del kernel (lo usaremos para los drivers de la nVidia y de la wifi).

Para instalarlos podríamos hacer, por ejemplo:

```
# apt-get -u install gcc libncurses5-dev linux-tree-2.6.21 kernel-package yaird module-assistant
```

También debemos quitar el paquete `initramfs-tools`. Esto es para que cuando creemos la imagen de arranque del sistema se use `yaird` en vez de este:

```
apt-get -u --purge initramfs-tools
```

Esto nos habrá dejado un archivo comprimido `tar.bz2` en el directorio `/usr/src` con el código fuente del kernel. Hay que descomprimirlo:

```
# cd /usr/src
# tar -xjf linux-source-2.6.21.tar.bz2
# ln -s linux-source-2.6.21/ linux
```

Con la última instrucción hemos creado un enlace simbólico al directorio donde está el código fuente del kernel. Esto es algo muy habitual y recomendable.

5.2. Configurando yaird

Como hemos dicho `yaird` va a crear una imagen reducida del kernel que se cargará en memoria en el arranque del sistema y realizará todo el proceso inicialización. Debemos configurar `yaird` para indicarle que módulos (drivers) debe incluir en esa imagen. Para ello editamos el fichero `/etc/yaird/Default.cfg`, y después de la línea:

```
MODULE mousedev
```

añadimos:

```
MODULE acpi-cpufreq
MODULE cpufreq_ondemand
MODULE ata_piix
MODULE sd_mod
```

donde:

- **acpi-cpufreq**: es quien se encarga de hacer el escalado de la frecuencia del procesador
- **cpufreq_ondemand**: es el "governor" de la frecuencia. Es decir, es quien decide cuando y como se va a escalar la frecuencia del procesador.
- **ata_piix** y **sd_mod**: dan soporte para acceder a nuestro disco SATA.

Con esto nos basta, ya que el resto de módulos se irán cargando de forma dinámica según sean necesarios.

5.2. Configuración del kernel

Lo primero es configurarlo, para ello ejecutamos:

```
# cd linux
# make menuconfig
```

Nos aparece un menú en la consola que nos permite configurar las opciones. Cabe destacar que a partir de la versión 2.6.20 el kernel ya trae opciones específicas para soportar los procesadores Core 2 Duo.

- Procesor type and features:

```
alex@pikkoro: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

Linux Kernel v2.6.21 Configuration

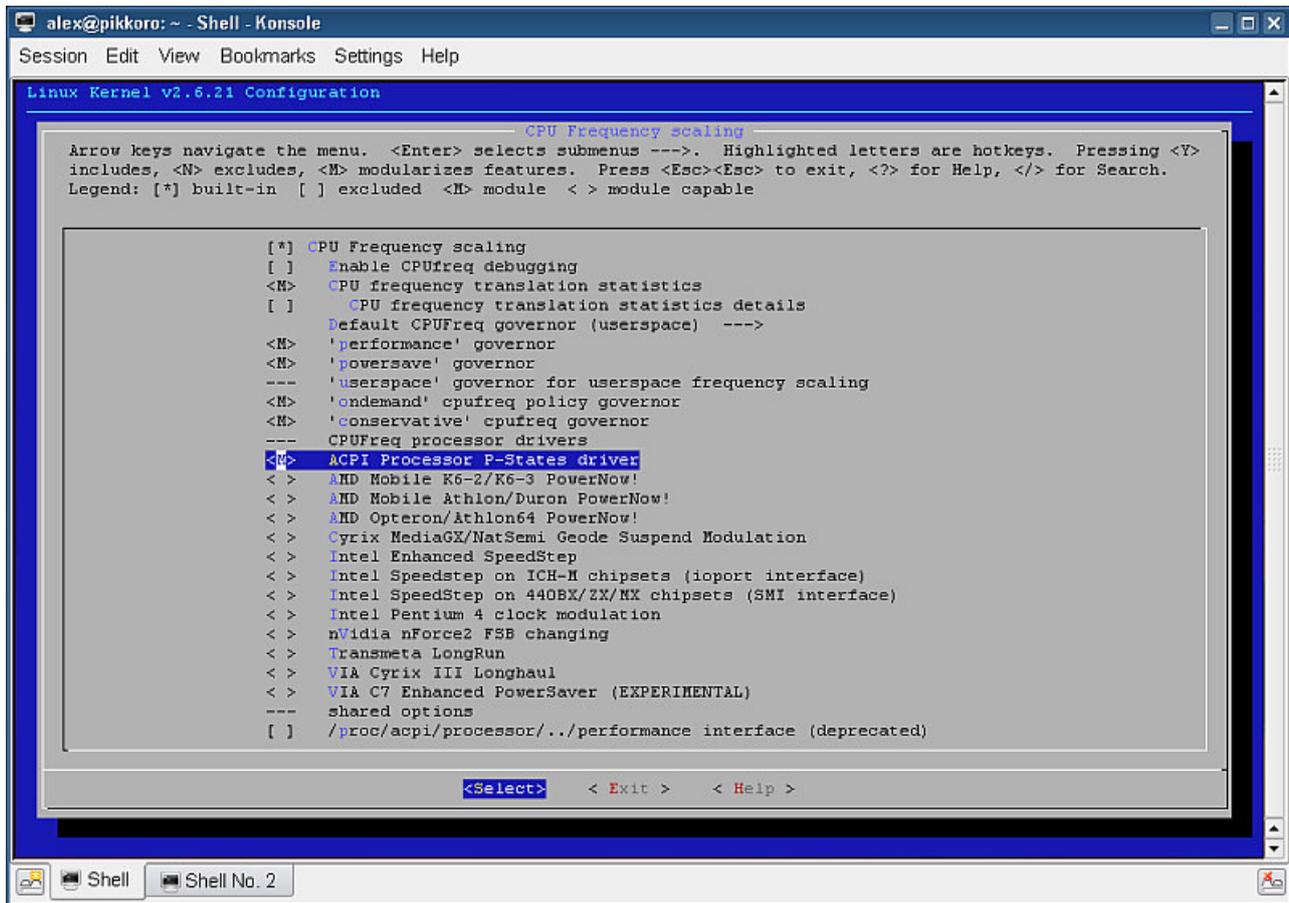
-- Processor type and features --
Arrow keys navigate the menu. <Enter> selects submenus ---. Highlighted letters are hotkeys. Pressing <Y>
includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search.
Legend: [*] built-in [ ] excluded <M> module < > module capable

[*] Tickless System (Dynamic Ticks)
[*] High Resolution Timer Support
[*] Symmetric multi-processing support
Subarchitecture Type (PC-compatible) --->
[ ] Paravirtualization support (EXPERIMENTAL)
[*] Processor family (Core 2/newer Xeon) --->
[ ] Generic x86 support
[*] HPET Timer Support
(2) Maximum number of CPUs (2-255)
[ ] SMT (Hyperthreading) scheduler support
[*] Multi-core scheduler support
Preemption Model (Voluntary Kernel Preemption (Desktop)) --->
[ ] Preempt The Big Kernel Lock
[*] Machine Check Exception
< > Check for non-fatal errors on AMD Athlon/Duron / Intel Pentium 4
[ ] check for P4 thermal throttling interrupt.
<M> Toshiba Laptop support
<M> Dell laptop support
[ ] Enable X86 board specific fixups for reboot
<M> /dev/cpu/microcode - Intel IA32 CPU microcode support
<M> /dev/cpu/*/msr - Model-specific register support
<M> /dev/cpu/*/cpuid - CPU information support
Firmware Drivers --->
High Memory Support (4GB) --->
Memory model (Sparse Memory) --->
--- Memory hotplug is currently incompatible with Software Suspend
v(+)

<Select> < Exit > < Help >
```

Se puede ver como la familia del procesador es "Core 2/newer Xeon", y que está activo la opción "Multi-core scheduler support". También hay que fijarse en que se ha deshabilitado al opción "SMT (Hyperthreading) scheduler support", y que el número de CPUs se ha fijado a 2 "(2) Maximum number of CPUs (2-255)".

- Power management options (ACPI, APM)
 - CPU Frequency scaling



En esta opción estamos preparando el soporte para el escalado dinámico del procesador. Es decir, a más demanda sube la frecuencia, y baja con menos demanda. Con esto conseguimos ahorrar energía, que se caliente menos, por lo tanto que haga menos ruido, y en definitiva alargar la vida de nuestro procesador.

Fijaros como hemos dado soporte para "ACPI Processor P-States driver", este es el driver adecuado para el Core 2 Duo.

Con esto hemos terminado el soporte para el Core 2 Duo, de todas formas [aquí](#) tenéis mi fichero de configuración por si queréis consultar que otras opciones tengo (se admiten recomendaciones ;).

5.3. Compilación del kernel

Para hacer la compilación ejecutaremos:

```
# make-kpkg --initrd --append-to-version .aca.070527 kernel-image
```

Con la opción `--initrd` estamos indicando que queremos generar un "disco en ram" para hacer el arranque del sistema. Es decir, se prepara un fichero con una imagen mínima del sistema, al arrancar el ordenador, carga en memoria esta imagen, y a partir de ella se carga el resto del sistema.

Para poder usar esta característica es necesario disponer de algún programa capaz de generar estas imágenes. Yo he elegido el `yaird`, que ya hemos comentado e instalado anteriormente.

El texto que ponemos detrás de la opción `--append-to-version` es simplemente algo que se va a concatenar al nombre del kernel que vamos a compilar para diferenciarlo fácilmente. A mi personalmente me gusta poner la fecha para saber cuando lo compilé, pero cada uno es libre de poner lo que quiera (si es que quiere poner algo ;)

`kernel-image` es el target que se encarga de compilar el kernel, compilar los módulos, y preparar un fichero `.deb` para instalar el nuevo kernel en nuestro sistema de forma sencilla.

5.4. Instalación del kernel

Para hacer la instalación del kernel compilado ejecutamos:

```
# cd ..
# dpkg -i linux-image-2.6.21.aca.070527_2.6.21.aca.070527-10.00.Custom_i386.deb
```

Donde `linux-image-2.6.21.aca.070527_2.6.21.aca.070527-10.00.Custom_i386.deb` es el nombre del fichero generado por `make-kpkg`.

Veremos como durante la instalación salen unos mensajes indicando que se ha modificado el GRUB para hacer referencia al nuevo kernel (lo habrá puesto el primero de la lista).

Ahora debemos reiniciar nuestro sistema y seleccionar el nuevo kernel durante el arranque.

6. Instalación del driver para la nVidia

Debemos instalar los paquetes:

- **nvidia-glx**
- **nvidia-kernel-common**
- **nvidia-kernel-source**
- **nvidia-settings**

Primero ejecutamos:

```
# apt-get -u install nvidia-kernel-common nvidia-kernel-source nvidia-settings
```

No instalamos todavía `nvidia-glx` ya que este paquete depende del driver y todavía no lo hemos compilado. Lo haremos más tarde.

Para compilar e instalar el driver hacemos:

```
# cd /usr/src
# module-assistant a-i nvidia-kernel
```

Esto se encarga de compilar el driver para el kernel que está actualmente en ejecución e instalar el driver.

Ahora ya podemos instalar `nvidia-glx`:

```
# apt-get -u install nvidia-glx
```

Ya sólo nos queda configurar las X para que usen el driver que acabamos de instalar. [Aquí](#) os adjunto el fichero de configuración `xorg.conf` para que veáis como hacerlo.

Ahora tenemos que reiniciar las X (o el sistema, lo que os resulte más cómodo).

7. Instalación del driver para la wireless

Nos descargamos el último firmware (microcode) de <http://bughost.org/ipw3945/>. En mi caso el fichero **ipw3945-ucode-1.14.2.tgz**. Lo descomprimos y copiamos el fichero `ipw3945.ucode` en `/lib/firmware`

```
# tar -xzf ipw3945-ucode-1.14.2.tgz
# cp ipw3945.ucode /lib/firmware
```

También podríamos haberlo descargado del repositorio de Debian (paquete `firmware-ipw3945`), pero yo he preferido descargarlo directamente de la web porque se trata de una versión más reciente (y en el tema del firmware nos interesa ;)

Ahora instalamos los paquetes:

- **ipw3945d**
- **ipw3945-source**

```
# apt-get -u install ipw3945d ipw3945-source
```

y compilamos el driver:

```
# cd /usr/src
# module-assistant a-i ipw3945
```

Igual que antes, `module-assistant` se encarga de compilar e instalar el driver.

Ya sólo tenemos que reiniciar la máquina para comprobar que todo arranca y se inicializa correctamente.

Además podemos comprobar que la tecla `Fn + F2` funciona perfectamente para habilitar o deshabilitar la wireless (puede resultar muy interesante si estamos con batería y queremos ahorrar un poquito más de energía).

8. Conclusiones

Actualmente instalar un GNU / Linux es tan fácil como instalar un Windows. En esta ocasión hemos visto Debian que tiene fama de ser de

las distribuciones más "ásperas", y sin embargo a sido muy sencillo. Incluso hemos visto como recompilar el kernel y como instalar algunos drivers para sacarle más partido a nuestro sistema.

Si a pesar de todo Debian os sigue dando "miedo" podéis intentarlo con Ubuntu (<http://www.ubuntu.com/>) o Kubuntu (<http://www.kubuntu.org/>). La única diferencia entre estas dos distribuciones es que Ubuntu instala GNOME por defecto, mientras que Kubuntu instala KDE, pero cualquiera de las dos tiene fama por su facilidad de instalación y configuración.

Lo que está claro es ya no tenemos excusa, por lo menos para probarlo. Y os puedo asegurar que una vez lo pruebas ya no lo cambias por nada. No sólo por que es open-source y ni el propio sistema ni las aplicaciones te cuestan un duro, sino porque podemos encontrar aplicaciones de mucha calidad (igual o mejor que lo que podemos encontrar en Windows):

- Openoffice (<http://www.openoffice.org/>) como suite de ofimática: escribir documentos, presentaciones, hojas de cálculo, ...
- Amarok (<http://amarok.kde.org/>) como reproductor multimedia (no encontraréis nada mejor).
- K3b (<http://k3b.plainblack.com/>) para grabar CDs, DVDs.
- Firefox (<http://www.mozilla.com/en-US/firefox/>) o Iceweasel (es la versión de Debian del Firefox) como navegador para Internet.
- Synaptic (<http://www.nongnu.org/synaptic/>) como gestor de paquetes. Esta es una de las características que más me gusta de Debian (y en general de Linux), que la instalación de prácticamente cualquier cosa, las aplicaciones la tienes centralizada en un sólo punto, y no como en Windwos que tienes que buscar la página en Internet, descargarte la aplicación, ... Además cuando actualizamos, no sólo se actualiza el sistema (como en Windows), sino que se nos actualizan todas las aplicaciones que tengamos instaladas.
- ...

Otro punto a favor de GNU / Linux es que no vamos a tener problemas de licencias. El uso de free-software y open-software se convierte en una alternativa más que viable a la piratería. NO a la piratería y NO a las patentes de software.

9. Sobre el autor

Alejandro Pérez García, Ingeniero en Informática (especialidad de Ingeniería del Software)

Socio fundador de Autentia (Formación, Consultoría, Desarrollo de sistemas transaccionales)

<mailto:alejandropg@autentia.com>

Autentia Real Business Solutions S.L. - "Soporte a Desarrollo"

<http://www.autentia.com>



This work is licensed under a [Creative Commons Attribution-NonCommercial-No Derivative Works 2.5 License](http://creativecommons.org/licenses/by-nc-nd/2.5/).
Puedes opinar sobre este tutorial [aquí](#)



Recuerda

que el personal de [Autentia](#) te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#))

¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?

¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

info@autentia.com

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos

Autentia = Soporte a Desarrollo & Formación

J2EE, EJBs, Struts...

[Autentia S.L.](#) Somos expertos en:
J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ..
y muchas otras cosas

Nuevo servicio de notificaciones

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales, inserta tu dirección de correo en el siguiente formulario.

| Subscribirse a Novedades | |
|--------------------------|---------------------------------------|
| e-mail | <input type="text"/> |
| | <input type="button" value="Enviar"/> |

Otros Tutoriales Recomendados ([También ver todos](#))

Nombre Corto

[Subversion, sistema de control de versiones, en Debian GNU/Linux](#)

[Wireless Centrino en Linux Debian](#)

[Emular XP en Debian con QEMU](#)

[Auto-gestión de dispositivos USB en Debian](#)

[Informar a Debian de la instalación de Java](#)

[Backup en Linux](#)

[Plugin PHPeclipse en Debian](#)

[Nombrar dispositivos USB en Debian](#)

[Instalación LAMP en Debian](#)

[Recordar primeros pasos por Linux](#)

Descripción

En esete tutorial aprenderemos a instalar y configurar el nuevo programa de gestión de versiones Subversion en Debian GNU/Linux

En este tutorial, Alejandro Perez nos enseña como configurar la red Wireless WIFI de tu centrino en Linux (Debian)

Alejandro Pérez no nos deja de sorprender con su conocimiento OpenSource. Hoy nos enseña a emular XP sobre Linux con QEMU.

En este tutorial y gracias a HAL (Hardware Abstraction Layer) vamos a ver como se puede gestionar de forma automática cualquier dispositivo que tengamos conectado en nuestro ordenador, discos duros, USB, firewire, ...

En este tutorial os mostramos como informar a Debian GNU/Linux de nuestra Máquina Virtual Java, no instalada desde paquetes Debian.

En este tutorial os mostramos cómo realizar copias de seguridad en un sistema Linux, utilizando sencillas herramientas del propio sistema operativo

Alejandro Pérez nos muestra como instalar el plugin de Eclipse: PHPeclipse, en GNU / Linux (Debian)

En este tutorial vamos a ver como podemos asignar nombres a los dispositivos USB en Debian, independientemente del nombre que le de el kernel.

Alejandro Perez nos enseña como instalar nuestro entorno Apache, MySQL y php en Debian

En este tutorial recordamos, de un modo visual, los primeros comandos en Linux sobre una emulación Qemu.

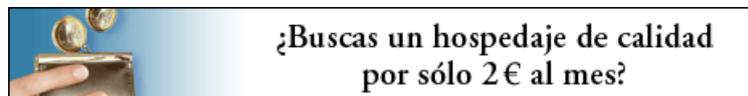
Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores.

En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo.

Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.

[Patrocinados por enredados.com Hosting en Castellano con soporte Java/J2EE](#)



www.AdictosAlTrabajo.com Optimizado 800X600