Cómo compilar el kernel Linux

Guillermo Valdez Lozano

13 de julio de 2007

Resumen

Este documento es una guía para la instalación del kernel Linux en una PC genérica. No pretende ser una referencia completa, por lo que su caso particular puede requerir de procedimientos no mostrados aquí.

Para adecuada comprensión de este documento se requiere tener nociones generales de GNU/Linux así como del uso de la consola. Si va efectuar alguna instalación en su equipo, haga respaldos de sus archivos importantes y tenga a la mano un disco de arranque de su distribución preferida.

Las distribuciones GNU/Linux Debian y Gentoo han sido usuadas para elaborar este material. Si usa otra distribución, el proceso pudiera ser algo diferente; aun así la mecánica fundamental debe ser la misma. No deje de revisar la documentación propia de su distribución.

A lo largo de este documento aparecen comandos que para diferenciar si deban ingresarse como superusuario o como usuario normal se antepone un $ si es ejecutado como usuario y # si es ejecutado como superusuario. Estos símbolos no forman parte del comando.

Agradezco de antemano su interés por leer este documento y deseo que le sea de utilidad.

Índice

1. Introducción

Linux es el núcleo o kernel del sistema operativo libre denominado GNU/Linux (también llamado Linux). Lanzado bajo la licencia pública general (GPL - General Public License) de GNU y desarrollado gracias a las contribuciones de todo el mundo. Linux es uno de los mejores ejemplos de software libre.

Linux fue creado por Linus Torvalds en 1991. El 25 de agosto de 1991, Torvalds escribió en comp.os.minix:

Estoy haciendo un sistema operativo (gratuito, sólo un hobby, no será nada grande ni profesional como GNU) para clones AT 386(486). Esto ha sido desde abril y está empezando a estar listo. Me gustaría un feedback de las cosas que les gustan o disgustan en minix. Actualmente he portado bash(1.08) y gcc(1.40), y parece que las cosas funcionan. Esto implica que tendré algo práctico en unos meses...

Tiene todas las características que usted espera de un kernel moderno, como multitarea, memoria virtual, librerías compartidas, carga a demanda o a petición, gestión apropiada de memoria y soporte para protocolos TCP/IP. Linux está protegido por la licencia GPL y está escrito en C.

1.1. ¿Qué es el kernel?

Para explicar en palabras sencillas lo que es un kernel, empecemos por saber que cada computadora, lector de discos, disco duro, tarjeta de video, etc. es diferente y el kernel es el programa que sabe que hacer para que esos componentes ejecuten su tarea como abrir un documento, borrarlo de un disco o mostrar una imagen en pantalla. El kernel debe saber cómo hablar a esos diferentes componentes materiales y controlar las labores corrientes que hacemos a diario con nuestras computadoras.

1.2. Versiones de Linux

Para darse una idea del crecimiento exponencial de Linux al paso de los años, basta con ver la cantidad de líneas de código que ha tenido en sus versiones:

Fecha	Versión	Cantidad de líneas de código
septiembre de 1991	0.01	10,239
14 de marzo de 1994	1.0.0	176,250
marzo de 1995	1.2.0	310,950
25 de enero de 1999	2.2.0	1,800,847
4 de enero de 2001	2.4.0	3,377,902
17 de diciembre de 2003	2.6.0	5,929,913

1.3. Kernel genérico vs kernel personalizado

Basándonos en la descripción anterior de lo que es un kernel, comprenderemos que cuando arrancamos un equipo con una distribución en CD-ROM o cuando tenemos una instalación reciente, estamos operando con un kernel genérico. Un kernel genérico es aquel capaz de funcionar en muchos tipos de equipos (digamos desde un Pentium II a un Pentium IV HT) y que tiene los controladores de la mayoría de los componentes que se pudiera encontar (como módulos que carga conforme se detectan los dispositivos).

Con un kernel compilado a la medida nos aseguramos que está optimizado al 100 % para nuestra máquina, añadimos los módulos que necesitamos y los que no necesitamos no se añaden (así ganamos rapidez en el arranque). Es de esperarse que un kernel personalizado es más ligero y más eficiente.

A todo lo anterior, agregue el hecho de que un kernel genérico suele ser de una *versión anterior* de Linux; por lo que crear su kernel personalizado le brinda la oportunidad de usar un kernel reciente que sea capaz de trabajar con nuevos componentes y que puede tener mejoras en su código, respecto a versiones anteriores.

2. Paquetes y descargas en Debian

La instalación inicial de Debian no incluye los paquetes necesarios para compilar un kernel (a diferencia de Gentoo). Como el superusuario instalamos los siguientes paquetes y las dependencias que soliciten.

```
$ su
# apt-get update
# apt-get install kernel-package
# apt-get install build-essential
```

```
# apt-get install linux-source-2.6.18
# apt-get install libncurses-dev
# apt-get install fakeroot
```

El kernel se depositará en el directorio **/usr/src** como un archivo comprimido. Lo desempacamos con:

```
# cd /usr/src
# tar xjf /usr/src/linux-source-2.6.18.tar.bz2
```

Le recomendamos que haga un enlace al directorio de las fuentes, para accesar por la ruta **/usr/src/linux**:

```
# ln -s linux-source-2.6.18 linux
```

3. Descargas en Gentoo

Como superusuario, actualizamos el *portage* para que el sistema conozca las versiones más recientes de los programas que pueda instalar.

```
$ su
# emerge --sync
```

En Gentoo están disponibles muchas variantes del kernel Linux. Puede hacer una búsqueda de los paquetes con la palabra *sources* con el siguiente comando:

```
# emerge -s sources
```

A continuación aparece una tabla con algunas de las opciones para la instalación del kernel y sus descripciones. Si no sabe cual elegir, use **gentoo-sources**.

Paquete	Descripción
gentoo-sources	Full sources including the Gentoo patchset for the 2.6 kernel tree
hardened-sources	Hardened kernel sources 2.6.18
suspend2-sources	Software Suspend 2 + Gentoo patchset sources
vanilla-sources	Full sources for the Linux kernel
xbox-sources	Full sources for the Xbox Linux kernel
xen-sources	Linux kernel 2.6.16 with Xen 3.0.2

Si ya ha compilado previamente un kernel, revise si existe una nueva versión para actualizar con:

```
# emerge -pu gentoo-sources
```

Si el comando anterior le informa que sí existe una nueva versión, borre el acceso directo **/usr/src/linux** y descarge el nuevo kernel:

```
# rm /usr/src/linux
# emerge -u gentoo-sources
```

Cuando va instalar por primera vez el kernel, simplemente ejecute:

```
# emerge gentoo-sources
```

Así tendrá descargadas y desempacadas las fuentes del kernel Linux en **/usr/src/linux**.

4. Tomar nota de los componentes del equipo

La mayor parte del trabajo para compilar un Kernel es el habilitar las opciones correctas en la interfaz de configuración. Por lo que es necesario conocer bien los componentes que forman el equipo.

4.1. Abrir el gabinete

La primer forma de obtener datos de los componentes del equipo es abrir el gabinete y tomar nota de las marcas, modelos y códigos de los chips en la tarjeta madre, en cada una de las tarjetas de expansión (PCI, ISA, etc), de los dispositivos de almacenamiento y escritura (discos duros, lectores de CD, DVD, etc).

Por ejemplo, en el equipo muestra se observaron los siguientes componentes:

- Tarjeta Madre Biostar P4TDP, Fury DDR 533
  - El disco duro y el CD-ROM se conectan por cintas IDE.
  - Tiene una ranura para tarjeta de video AGP.
  - Chip VIA VT6202 0208CD.
  - Chip de audio integrado C3DX CMI 8738/PCI-6ch-LX (pero como tengo una tarjeta de audio prefiero desactivarlo en el BIOS).
  - Chip ATA 100.
  - Tiene puertos USB por delante y por detrás.
- Tarjeta de Video AGP de 32 MB con etiqueta TAVCATG1293A
- Tarjeta de Audio Genius con chip C3DX CMI 8738/PCI-6ch-LX
- Tarjeta de Red con chip DM9102AF

4.2. Comando lspci

Nuestra segunda fuente de información es el comando **lspci** el cual nos lista los dispositivos PCI. En el equipo muestra nos entrega:

```
$ lspci
00:00.0 Host bridge: Intel Corporation 82845 845 (Brookdale)
        Chipset Host Bridge (rev 04)
00:01.0 PCI bridge: Intel Corporation 82845 845 (Brookdale)
        Chipset AGP Bridge (rev 04)
00:1e.0 PCI bridge: Intel Corporation 82801 PCI Bridge (rev 05)
00:1f.0 ISA bridge: Intel Corporation 82801BA ISA Bridge (LPC) (rev 05)
00:1f.1 IDE interface: Intel Corporation 82801BA IDE U100 (rev 05)
```

```
00:1f.2 USB Controller: Intel Corporation 82801BA/BAM USB (Hub #1) (rev 05)
00:1f.3 SMBus: Intel Corporation 82801BA/BAM SMBus (rev 05)
00:1f.4 USB Controller: Intel Corporation 82801BA/BAM USB (Hub #2) (rev 05)
01:00.0 VGA compatible controller: Silicon Integrated Systems
        [SiS] 300/305 PCI/AGP VGA Display Adapter (rev 90)
02:00.0 Multimedia audio controller: C-Media Electronics Inc CM8738 (rev 10)
02:04.0 Ethernet controller: Davicom Semiconductor, Inc. 21x4x
        DEC-Tulip compatible 10/100 Ethernet (rev 31)
02:05.0 USB Controller: VIA Technologies, Inc. VT82xxxxx
        UHCI USB 1.1 Controller (rev 50)
02:05.1 USB Controller: VIA Technologies, Inc. VT82xxxxx
        UHCI USB 1.1 Controller (rev 50)
02:05.2 USB Controller: VIA Technologies, Inc. USB 2.0 (rev 51)
```

Analizando los datos tomados al abrir el equipo y los entregados por lspci podemos concluir lo siguiente:

- En la tarjeta madre tenemos...
  - Intel 82845 Host Brige, AGP Brige
  - Intel 82801 PCI Brige
  - Intel 82801BA ISA Brige, IDE U100, USB, SMBus
  - VIA Tech 82xxxxx
- En la tarjeta de video...
  - SIS 300/305 VGA
- En la tarjeta de audio...
  - C-Media CMI 8738
- Y en la tarjeta de red...
  - Davicom DEC-Tulip Ethernet

4.3. Observar los módulos cargados

La tercer fuente de información es observar los módulos que se hayan cargado automáticamente por el kernel genérico. Con el comando **lsmod**.

```
# lsmod
```

Se requiere conocer bastante bien al kernel para saber (o adivinar :D) lo que hace cada módulo. De este largo listado podemos destacar:

Módulo	**Descripción**
sis	Controlador de la tarjeta de video SIS
partport_pc	Puerto paralelo
gameport	Hay un puerto de juegos/midi
floppy	Controlador del lector de floppys
snd_cmipci	Controlador de la tarjeta de sonido Genius
i2c_i801	Tiene que ver con un chip intel
psmouse	Mouse por el puerto PS/2
intel_agp	Ranura AGP de la tarjeta madre
usbhid, ehci_hcd, uhci_hcd	Tienen que ver con el USB
ide_cd, ide_disk	Controladores del CD-ROM y del disco duro
dmfe	Controlador de la tarjeta de red Tulip/Davicom

4.4. Bitácora del kernel

Como cuarta fuente de información podemos ver la bitácora de arranque del kernel con **dmesg**. Este comando nos entrega un muy largo listado, usted no está obligado a verlo, pero si lo hace podrá encontrar detalles adicionales de las características de su equipo.

```
# dmesg | more
```

En el equipo muestra podemos notar que tiene capacidades de *Plug And Play*:

```
Linux Plug and Play Support v0.97 (c) Adam Belay
pnp: PnP ACPI: disabled
PnPBIOS: Scanning system for PnP BIOS support...
PnPBIOS: Found PnP BIOS installation structure at 0xc00fbb10
PnPBIOS: PnP BIOS version 1.0, entry 0xf0000:0xbb40, dseg 0xf0000
PnPBIOS: 14 nodes reported by PnP BIOS; 14 recorded by driver
isapnp: Scanning for PnP cards...
isapnp: No Plug & Play device found
```

Reconoce las capacidades ICH en la tarjeta madre:

```
ICH2: IDE controller at PCI slot 0000:00:1f.1
ICH2: chipset revision 5
ICH2: not 100% native mode: will probe irqs later
ide0: BM-DMA at 0xf000-0xf007, BIOS settings: hda:DMA, hdb:pio
ide1: BM-DMA at 0xf008-0xf00f, BIOS settings: hdc:DMA, hdd:DMA
```

Se detecta el disco duro y las unidades CD-RW y DVD.

```
hda: IC35L120AVV207-0, ATA DISK drive
ide0 at 0x1f0-0x1f7,0x3f6 on irq 14
Probing IDE interface ide1...
hdc: HL-DT-ST GCE-8520B, ATAPI CD/DVD-ROM drive
hdd: HL-DT-ST DVDRAM GSA-H10A, ATAPI CD/DVD-ROM drive
ide1 at 0x170-0x177,0x376 on irq 15
hda: max request size: 512KiB
hda: 241254720 sectors (123522 MB) w/1821KiB Cache,
     CHS=16383/255/63, UDMA(100)
hda: cache flushes supported
hda: hda1 hda2 hda3 hda4 < hda5 hda6 hda7 hda8 hda9 hda10 >
hdc: ATAPI 40X CD-ROM CD-R/RW drive, 2048kB Cache, UDMA(33)
Uniform CD-ROM driver Revision: 3.20
hdd: ATAPI 48X DVD-ROM DVD-R-RAM CD-R/RW drive,
     2048kB Cache, UDMA(33)
```

Se detecta la ranura AGP de la tarjeta de video.

```
Linux agpgart interface v0.101 (c) Dave Jones
agpgart: Detected an Intel i845 Chipset.
agpgart: AGP aperture is 64M @ 0xe8000000
```

Vemos la carga del módulo que controla la tarjeta de red.

```
dmfe: Davicom DM9xxx net driver, version 1.36.4 (2002-01-17)
```

4.5. Procesador

También debe saber el modelo del procesador de su equipo, fácilmente este comando le mostrará información detallada del mismo:

```
# cat /proc/cpuinfo
processor     : 0
vendor_id     : GenuineIntel
cpu family    : 15
model         : 1
model name    : Intel(R) Pentium(R) 4 CPU 1.70GHz
stepping      : 2
cpu MHz       : 1700.335
cache size    : 256 KB
fdiv_bug      : no
hlt_bug       : no
f00f_bug      : no
coma_bug      : no
fpu           : yes
fpu_exception : yes
cpuid level   : 2
wp            : yes
flags         : fpu vme de pse tsc msr pae mce cx8 apic sep
                mtrr pge mca cmov pat pse36 clflush dts acpi
                mmx fxsr sse sse2 ss ht tm up
bogomips      : 3404.16
```

5. Configurar

5.1. Interfaz de configuración

Teniendo a la mano toda la infomación recabada, sigue la configuración del kernel. Este paso consiste en activar cada opción que requiera nuestro equipo con el fin de que forme parte del nuevo kernel. Como superusuario, cámbiese al directorio de las fuentes de Linux:

```
# cd /usr/src/linux
```

Para que se establezca la configuración por defecto del kernel, ejecute **make defconfig**. Este comando activa algunas opciones de uso común, pero no es el trabajo de configuración terminado, sólo una sugerencia para comenzar:

```
# make defconfig
```

Para configurar en la misma consola, usando la interfaz *ncurses* (opción recomendada) ejecutamos:

```
# make menuconfig
```

Si tiene instalda las X's y las librerias GTK, puede configurar el kernel también con:

```
# make gconfig
```

Si tiene KDE e instaladas las librerías QT necesarias, puede configurar el kernel con:

```
# make xconfig
```

5.2. Uso de la interfaz de configuración bajo *ncurses*

Al ejecutar **make menuconfig** tendrá en pantalla el menú raiz de la configuración.

Use las fechas del teclado para navegar en el menú, puede saltar de inmediato a una opción tecleando la letra resaltada. Con **Enter** podrá entrar a un submenú. Estando sobre una opción presione **Y** para incluirla dentro del kernel, **N** para excluirla o **M** para que sea un módulo.

Cada opción tiene a su izquierda una zona que indica cómo esta configurada, si es `[*]` significa que será embebida en el kernel, si es `[ ]` no será incluida; cuando se solicita que sea un módulo aparece como `<M>` y cuando tiene capacidad de ser un módulo pero no será compilada, será `<>`.

Si necesita ayuda presione **?**, verá un texto descriptivo de la opción donde se encuentre. Para regresar al menú anterior presione **ESC** o elija la opción *Exit* de la parte inferior.

Para salir guardando cambios, estando en el menu raíz presione **ESC** e indique guardar. Para salir sin guardar cambios, desde el menú raíz presione **ESC** dos veces.

5.3. Kernel monolítico vs kernel modular

Desde un punto de vista extremista, un kernel monolítico tiene todas las opciones marcadas con **Y**, mientras que un kernel modular es aquel con la mayoría de las opciones marcadas con **M**[1].

Se recomienda un kernel de forma monolítica cuando el hardware no cambia, por ejemplo, para una computadora portátil donde sus dispositivos están incorporados, no se desconectan o ni se deshabilitan. En cambio, si tiene dispositivos que no están conectados de forma permanente, como su agenda electrónica o el control de juegos, conviene manejarlos como módulos.

Un kernel de modo modular le facilita la experimentación de los dispositivos, ya puede cargar a voluntad los módulos que necesite, por ejemplo, si tiene dos tarjetas de audio distintas y sólo desea activar una, cargue los módulos de la elegida con el comando **modprobe**.

Tenga cuidado de no marcar como módulos aquellas opciones que sean indispensables en el arranque, por ejemplo, para un servidor con discos *SCSI* el kernel debe tener embebido el controlador *SCSI* (como Y); de lo contrario, si estubiera como módulo, le sería imposible arrancar el disco duro para cargar el sistema operativo.

5.4. Cambios del menú de configuración en las versiones del kernel

Si va a instalar una versión nueva comparada a la que tiene en uso, usted esperará que tendrá código mejorado, por ejemplo, tal vez aparezca la más reciente versión del controlador de su tarjeta de red inalámbrica. Por esto, siempre es bueno tener la versión más reciente del kernel Linux.

El constante crecimiento y mejora del kernel Linux ocasiona que el menú de configuración cambie de una versión a otra. En algunas ocasiones, partes del menu se mueven, es decir, son colocadas en otras ramas; por ejemplo, a partir de la versión 2.6.20 todos los controladores *SATA* se encuentran en la rama *Serial ATA (prod) and Parallel ATA (experimental) drivers* la cual no encontrará en versiones anteriores a la 2.6.20.

También ocurre el aviso y luego el retiro de código en el kernel, esto es, que código anterior puede ser abandonado y que desaparecerá en una versión futura. Por ejemplo, *Open Sound System* que es el anterior sistema de sonido, tiene la leyenda *(DEPRECATED)* que significa que será suprimido en un futuro; si lo necesita puede activarlo, pero no se le recomienda.

[1]No es posible que todas las opciones sean modulares, ya que muchas de éstas no lo permiten

ADVERTENCIA: Los ejemplos del menú de configuración del kernel de este documento son de las versiones 2.6.18 y 2.6.20. Tenga en cuenta que pueden ser diferentes las opciones si usa otra versión.

5.5. Configuraciones básicas

Active la opción *Prompt for development and/or incomplete code/drivers* para que aparezcan las opciones de tipo experimental, entre las cuales están los controladores de los componentes más recientes:

```
Code maturity level options  --->
  [*] Prompt for development and/or incomplete code/drivers
```

Si piensa habilitar algunas opciones como módulos, le puede resultar benéfica la opción de carga automática de módulos; así cuando conecte un dispositivo externo (por ejemplo, el control de video juegos *USB*) el módulo se cargará por sí sólo:

```
Loadable module support  --->
  [*] Enable loadable module support
  [*]   Module unloading
  [*]     Forced module unloading
  [*]   Automatic kernel module loading
```

Seleccione la familia correcta a la que pertenece el procesador. Por ejemplo, para un *Pentium* 4:

```
Processor type and features  --->
  Processor family (Pentium-4/Celeron(P4-based)/Pentium-4 M/Xeon)
```

Si tiene un procesador *Intel* con capacidad *HyperThreading* (HT), o bien, con dos o más procesadores, debe activar la opción de soporte de multi-procesamiento simétrico:

```
Processor type and features  --->
  [*] Symmetric multi-processing support
```

Recuerde que todos los controladores que sean indispensables para el arranque **NO** deben ser seleccionados como módulos, entre ellos están los sistemas de archivos. Marque con **Y** los sistemas de archivos de las particiones del disco duro que deban montarse al arranque, por ejemplo, si tiene particiones formateadas en *Ext2* y *Ext3*:

```
File systems  --->
  <*> Second extended fs support
  [*]   Ext2 extended attributes
  <*> Ext3 journalling file system support
  [*]   Ext3 extended attributes
```

Active *inotify* para que los administradores de archivos muestren los cambios sin necesidad de refrescar y los pseudos-sistemas *proc* y de memoria virtual:

```
File systems  --->
  [*] Inotify file change notification support
  [*]   Inotify support for userspace
  Pseudo filesystems  --->
    --- /proc file system support
    [*]   /proc/kcore support
    [*] Virtual memory file system support (former shm fs)
```

Seguramente su equipo estará conectado a una red local o a internet, lo cual requiere habilitar las capacidades básicas de red. Este apartado es extenso y debe ser cuidadosamente configurado si el fin del equipo es ser un *muro de fuego* o un *ruteador*. Las opciones comunes para simplemente tener acceso a una red son:

```
Networking  --->
  [*] Networking support
    Networking options  --->
      <*> Packet socket
      [*]   Packet socket: mmapped IO
      <*> Unix domain sockets
      [*] TCP/IP networking
      [*] Network packet filtering (replaces ipchains)  --->
        Core Netfilter Configuration  --->
          <*> Netfilter Xtables support (required for ip_tables)
          <*>   "limit" match support
          <*>   "mac" address match support
```

```
  <*>   "state" match support
IP: Netfilter Configuration  --->
  <*> Connection tracking (required for masq/NAT)
  <*>   FTP protocol support
  <*> IP tables support (required for filtering/masq/NAT)
  <*>   Packet filtering
```

5.6. Configuraciones para dispositivos de almacenamiento y USB

Es recomendable activar las siguientes opciones del apartado *Block devices*: Soporte para discos *floppys*, capacidad para montar imágenes de disco ISO en un direcctorio con *loopback*, capacidad de operar discos virtuales en RAM y capacidad de escritura como paquetes para quemadores de CD/DVD.

```
Device Drivers  --->
  Block devices  --->
  <*> Normal floppy disk support
  <*> Loopback device support
  <*> RAM disk support
  [*] Initial RAM filesystem and RAM disk (initramfs/initrd) suppor
  <*> Packet writing on CD/DVD media
```

Si sus discos duros y unidades de disco se conectan por cintas *IDE*, active las opciones *ATA/ATAPI* y el chip que lo controla. En el siguiente ejemplo se tiene un chip *Intel* en la tarjeta madre y con capacidad *DMA*:

```
Device Drivers  --->
  ATA/ATAPI/MFM/RLL support  --->
  <*> ATA/ATAPI/MFM/RLL support
  <*>   Enhanced IDE/MFM/RLL disk/cdrom/tape/floppy support
  <*>     Include IDE/ATA-2 DISK support
  <*>     Include IDE/ATAPI CDROM support
  <*>     Include IDE/ATAPI FLOPPY support
  <*>     generic/default IDE chipset support
  [*]     PCI IDE chipset support
  [*]       Sharing PCI IDE interrupts support
  <*>       Generic PCI IDE Chipset Support
  [*]       Generic PCI bus-master DMA support
```

```
  [*]         Use PCI DMA by default when available
  <*>         Intel PIIXn chipsets support
```

Si la tarjeta madre usa cables *SATA*, active el soporte *Serial ATA* y marque el controlador necesario, en el siguiente ejemplo, para un chip *Intel*:

```
Device Drivers  --->
  Serial ATA (prod) and Parallel ATA (experimental) drivers  --->
  <*> ATA device support
  <*>   Intel PIIX/ICH SATA support
```

Algo indispensable para el montaje de los dispositivos de almacenamiento que conecte vía *USB* (por ejemplo una memoria USB) y para los quemadores de CD y DVD es el soporte *SCSI*. Parece extraño, pero actívelo aunque NO cuente con dispositivos *SCSI*:

```
Device Drivers  --->
  SCSI device support  --->
    [*] legacy /proc/scsi/ support
    <*> SCSI disk support
    <*> SCSI CDROM support
    <*> SCSI generic support
```

Si tiene dispositivos *USB* (como teclado, ratón, memorias o discos duros externos) no olvide activar el soporte para los mismos. En este ejemplo están habilitados como módulos:

```
Device Drivers  --->
  USB support  --->
    <*> Support for Host-side USB
    [*]   USB device filesystem
    <M>   EHCI HCD (USB 2.0) support
    [*]     Full speed ISO transactions (EXPERIMENTAL)
    [*]     Root Hub Transaction Translators (EXPERIMENTAL)
    <M>   OHCI HCD support
    <M>   UHCI HCD (most Intel and VIA) support
    <M>   USB Printer support
    <*> USB Mass Storage support
    <M> USB Human Interface Device (full HID) support
    [*]   HID input layer support
```

5.7. Configuración para la tarjeta de red

Marque el controlador adecuado de su tarjeta de red alámbrica. Para el ejemplo de la tarjeta Davicom, esta es la configuración:

```
Device Drivers  --->
  Network device support  --->
    [*] Network device support
        Ethernet (10 or 100Mbit)  --->
          Tulip family network device support  --->
            [*] "Tulip" family network device support
            <*>   Davicom DM910x/DM980x support
```

Comparado a una tarjeta de red alámbrica, una inalámbrica requiere que se activen más opciones. Esta es la configuración para una tarjeta de red inalábrica *Intel PRO/Wireless 2200BG* con soporte para cifrado *WEP*:

```
Networking  --->
  --- Networking support
  <M>   Generic IEEE 802.11 Networking Stack
  <M>     IEEE 802.11 WEP encryption (802.1x)
  <M>     IEEE 802.11i CCMP support
  <M>     IEEE 802.11i TKIP encryption
Device Drivers  --->
  Network device support  --->
    [*] Network device support
    Wireless LAN (non-hamradio)  --->
      [*] Wireless LAN drivers (non-hamradio) & Wireless Extensions
      <M>   Intel PRO/Wireless 2200BG and 2915ABG Network Connection
      [*]     Enable promiscuous mode
      [*]     Enable QoS support
Cryptographic options  --->
  --- Cryptographic API
  <M>   Cryptographic algorithm manager
  <M>   SHA1 digest algorithm
  <M>   SHA256 digest algorithm
  <M>   ECB support
  <M>   CBC support
  <M>   AES cipher algorithms
  <M>   AES cipher algorithms (i586)
```

```
  <M>   ARC4 cipher algorithm
  <M>   Michael MIC keyed digest algorithm
```

5.8. Configuración para la tarjeta de video

La tarjeta de video es una pieza importante de la configuración del kernel, sobre todo si necesita aprovechar sus capacidades de aceleración en las *X*. Como ejemplo, para usar los controladores libres *ATI* en una tarjeta de la familia *Radeon*:

```
Device Drivers  --->
  Character devices  --->
    <*> /dev/agpgart (AGP Support)
    <M>   ATI chipset support
    <*> Direct Rendering Manager (XFree86 4.1.0 and higher DRI support)
    <M>   ATI Radeon
    [*] HPET - High Precision Event Timer
    [*]   Allow mmap of HPET
    <*> Hangcheck timer
```

5.9. Configuración para la tarjeta de audio

Como se comentó con anterioridad, el *Open Sound System* (OSS) está marcado como depreciado, lo que significa que será removido en una versión futura del kernel. Para habilitar la tarjeta de sonido marque como módulos las opciones de *Advanced Linux Sound Architecture* (ALSA) y no olvide también marcar como módulo el controlador de la tarjeta misma, en este ejemplo, vemos habilitada la tarjeta C-Media:

```
Device Drivers  --->
  Sound  --->
    <*> Sound card support
    Advanced Linux Sound Architecture  --->
      <M> Advanced Linux Sound Architecture
      <M>   Sequencer support
      <M>   OSS Mixer API
      <M>   OSS PCM (digital audio) API
      [*]     OSS PCM (digital audio) API - Include plugin system
      [*]   OSS Sequencer API
```

```
  <M>   RTC Timer support
  [*]     Use RTC as default sequencer timer
  [*]   Support old ALSA API
  PCI devices  --->
    <M> C-Media 8738, 8338
Open Sound System  --->
  < > Open Sound System (DEPRECATED)
```

5.10. Guardar la configuración

Cuando termine de configurar no deje de revisar lo que ha hecho, porque puede ocurrir que una simple omisión haga que falle el kernel cuando lo estemos probando por primera vez. De cualquier modo, no se preocupe, como verá más adelante es recomendable mantener algunas versiones probadas que le permitan reiniciar y volver a configurar, compilar e instalar el kernel que está preparando.

Para salir del menu de configuración presione **ESC** estando en la raíz, o bien, indique la opción *Exit*; le preguntará si quiere guardar los cambios, a lo que seguramente debe responder que sí.

Toda la configuración queda guardada en un archivo oculto de texto llamado **.config** el cual puede respaldar (o modificar) si desea hacerlo.

6. Compilar en Debian

En Debian el kernel se compila y se empaqueta para crear un sólo archivo **.deb** que contiene el kernel mismo y los módulos estándard habilitados.

Tenemos dos alternativas al compilar, crear un kernel *sin initrd* o *con initrd*. El disco RAM inicial (*initrd*) es un sistema de archivos temporal usado por el kernel durante el arranque. Se usa para hacer preparaciones iniciales antes de que la verdadera partición raíz (el directorio /) sea montada. Algún requerimiento particular pudiera necesitar que sea creado *con initrd*, como por ejemplo, para que una portátil pueda *suspenderse*[2] e *hibernarse*[3].

Como preparativo en Debian, puede solicitar que se eliminen las compilaciones hechas con anterioridad con:

[2]Suspender apaga el monitor y mantiene datos en la RAM consumiendo poca energía.
[3]Hibernar vacia el contenido de la RAM al disco duro y apaga el equipo.

```
# cd /usr/src/linux
# make-kpkg clean
```

Para crear un kernel *sin initrd*, ejecute el comando **fakeroot**, donde puede cambiar el texto *personalizado.1.0* por un nombre y número de versión que le sirva a usted de referencia:

```
# fakeroot make-kpkg --revision=personalizado.1.0 kernel_image
```

Para crear un kernel *con initrd*, sólo debe agregar el parámetro –**initrd**; ejecute:

```
# fakeroot make-kpkg --initrd --revision=personalizado.1.0 kernel_image
```

Y puede dejar el equipo compilando para irse a tomar un café, por varios minutos :-). Al terminar la compilación, revise que se haya creado el paquete debian en el directorio **/usr/src**:

```
# ls -l /usr/src
```

7. Instalar el kernel en Debian

No se recomienda que instale una versión del kernel igual a la que esté usando en ese momento, porque podrían eliminarse algún módulo que pueda necesitarse. Mejor reinicie y elija un kernel distinto para luego instalar el nuevo.

Ejecute el comando **dpkg** para instalar, cambie el nombre del archivo **.deb** por el que corresponda al kernel que haya elaborado:

```
# cd /usr/src
# dpkg -i linux-image-2.6.18_personalizado.1.0_i386.deb
```

Este comando hará por usted todos estos pasos:

- Instalará en **/boot** el nuevo kernel junto con el *initrd* si fue solicitado.

- Copiará los módulos al directorio **/lib/modules/*version-del-kernel***
- Agregará la opción para seleccionarlo en el gestor de arranque *Grub o Lilo*.

En este punto ya tendrá listo su nuevo kernel. Así que reinicie su equipo y elíjalo del menú de su gestor de arranque.

8. Compilar en Gentoo

En Gentoo Linux se hace la compilación al estilo tradicional. Si comparamos el proceso de compilar e instalar entre Debian y Gentoo, encontrará que en Gentoo es un procedimiento con más pasos, pero a la vez es más intuitivo y controlable.

Como preparativo, si ya ha hecho una compilación previa, puede hacer una *limpia* con el comando:

```
# make clean
```

En mi experiencia en Gentoo no he necesitado compilar el kernel con la solicitud de crear un *initrd*. Si ya tiene configurado su kernel, para compilarlo simplemente ejecute:

```
# make
```

Le sugerimos que tome un descanso en lo que su equipo compila :D.

9. Instalar el kernel en Gentoo

9.1. Copiar el kernel a /boot

Después de la compilación el kernel será depositado en **arch/i386/boot/bzImage**. Monte la partición **/boot** y copie el kernel a la misma. Puede modificar el nombre del archivo de destino **kernel-2.6.20-r8** a su gusto; tome nota de éste nombre porque se necesitará al modificar la lista del gestor de arranque:

```
# mount /boot
# cp arch/i386/boot/bzImage /boot/kernel-2.6.20-r8
```

Una buena sugerencia es hacer una copia de la configuración del kernel al mismo destino **/boot**, ésto sólo con fines de respaldo.

```
# cp .config /boot/config-2.6.20-r8
```

9.2. Instalar los módulos

El siguiente paso es instalar los módulos compilados (salvo si no marcó ninguna opción con **M**, puede saltarse esta sección).

```
# make modules_install
```

Si desea que se carguen siempre algunos módulos en el arranque, agregue los nombres de éstos y sus parámetros en el archivo **/etc/modules.autoload.d/kernel-2.6** y luego ejecute **update-modules**.

```
# nano -w /etc/modules.autoload.d/kernel-2.6
# update-modules
```

Como ejemplo, este es el contenido del archivo **/etc/modules.autoload.d/kernel-2.6** en mi computadora portátil:

```
# CPU Frequency Utils para ACPI
acpi-cpufreq

# Tarjeta de video ATI Radeon
radeon

# Tarjeta inalambrica de red
ipw2200
```

Este es otro ejemplo del archivo **/etc/modules.autoload.d/kernel-2.6** donde se pasan parámetros a los módulos:

```
# Tarjeta de TV Pinnacle 110i
#   card=77  -> Pinnacle PCTV 40i/50i/110i (saa7133)
#   tuner=61 -> Tena TNF9533-D/IF/TNF9533-B/DF
saa7134      card=77 tuner=61
saa7134-alsa index=2,3
```

9.3. Modificar el gestor de arranque GRUB

Modifique el gestor de arranque con el comando:

```
# nano -w /boot/grub/menu.lst
```

Como ejemplo, las siguientes líneas indican que la partición *booteable* es la primera del primer disco duro, que el kernel es el archivo **kernel-2.6.20-r8** y que la partición raíz del sistema operativo linux está en la tercera partición del disco duro:

```
title  Mi kernel Linux personalizado 2.6.20-r8
root   (hd0,0)
kernel /kernel-2.6.20-r8 root=/dev/hda3
```

Este otro ejemplo es el de mi computadora portátil, donde se ve un parámetro más al kernel, *resume* se refiere a la partición *swap* donde puede guardar los datos de la memoria para poder *hibernar* el equipo:

```
title  Gentoo Linux - 2.6.20-r8
root   (hd0,0)
kernel /gentoo-kernel-2.6.20-r8 root=/dev/sda9 resume=/dev/sda2
```

¡Listo! Puede reiniciar su equipo y probar el nuevo kernel.

9.4. Solución de errores y revisión del nuevo kernel

Sin duda lo más estresante es el arranque de nuevo kernel. La probabilidad de que funcione perfectamente al primer arranque es poca. Es normal que tenga que revisar de nuevo la configuración y volver a compilar, instalar y probar.

Si hubo un error fatal en el arranque, tome nota de los mensajes en su pantalla, reinicie y carge el kernel anterior. Vuelva a ejecutar el comando **make menuconfig** para que revise las configuraciones vitales como el tipo de procesador, sistemas de archivos, dispositivos ATA o SATA.

Si el arranque es satisfactorio, no deje de revisar:

- La bitácora con el comando **dmesg — more**, busque particularmente mensajes de error.
- Que se hayan cargado los módulos que necesite con el comando **lsmod**.
- Revise que funcionen correctamente los principales dispositivos (video, red, audio, teclado, ratón).
- Pruebe los dispositivos que no estén siempre conectados al equipo, como memorias USB.

Para muestra *basta un botón*, la tarjeta inalámbrica de mi portátil (Intel ipw2200bg) requiere que habilite muchas opciones en la configuración, pero con todo eso revisado una y otra vez, no conseguía que funcionara, hasta que noté el siguiente mensaje con el comando **dmesg — more**:

```
ipw2200: Intel(R) PRO/Wireless 2200/2915 Network Driver, 1.2.0kmq
ipw2200: Copyright(c) 2003-2006 Intel Corporation
ipw2200: Detected Intel PRO/Wireless 2200BG Network Connection
ipw2200: ipw2200-bss.fw request_firmware failed: Reason -2
ipw2200: Unable to load firmware: -2
ipw2200: failed to register network device
ipw2200: probe of 0000:03:03.0 failed with error -5
```

Como podrá ver, marca un error porque no encuentra el *firmware*. Al leer la documentación del controlador en **/usr/src/linux/Documentation/networking/README.ipw2200** encontré esto:

```
5.  Firmware installation
-----------------------------------------------

The driver requires a firmware image, download it and extract the
```

```
files under /lib/firmware (or wherever your hotplug's firmware.agent
will look for firmware files)

The firmware can be downloaded from the following URL:

    http://ipw2200.sf.net/
```

Con lo que aprendí que no bastaba con habilitar el controlador en la configuración del kernel; también tenía que descargar e instalar los archivos del *firmware*:

```
$ tar xvf ipw2200-fw-3.0.tgz
$ su
# mkdir /lib/firmware
# cp * /lib/firmware/
```

Al reiniciar, la carga del controlador de la tarjeta de red inalámbrica fue exitosa:

```
$ dmesg | grep -i ipw2200
ipw2200: Intel(R) PRO/Wireless 2200/2915 Network Driver, 1.2.0kmq
ipw2200: Copyright(c) 2003-2006 Intel Corporation
ipw2200: Detected Intel PRO/Wireless 2200BG Network Connection
ipw2200: Detected geography ZZM (11 802.11bg channels, 0 802.11a channels)
```

10. Eliminar un kernel anterior

Es muy estimulante que al pertenecer a la comunidad del Software Libre gozemos de actualizaciones frecuentes de cualquier aplicación del GNU/Linux. Al paso del tiempo habremos dejado de usar versiones anteriores y podemos decidir eliminarlas de nuestro disco duro para ganar espacio en el mismo.

Antes de empezar a eliminar versiones anteriores, tenga en cuenta que puede conservar el archivo **.config**. Así si por algún motivo necesitara reinstalar una versión anterior, sólo copie el archivo de configuración como **.config** en el directorio de las fuentes del kernel y lo tendrá listo para compilarlo de nuevo.

10.1. Remover un kernel en Debian

Si instaló el kernel en Debian a partir de un archivo .deb lo desinstala con el comando **dpkg -r *nombre-del-paquete***:

```
# dpkg -r linux-image-2.6.17
```

10.2. Remover un kernel en Gentoo

Para remover un kernel el Gentoo debemos montar la partición /boot y eliminar el archivo del kernel:

```
# mount /boot
# rm /boot/kernel-2.6.17-r4
```

Eliminar las líneas que lo definien en la lista de opciones del gestor de arranque GRUB, editando el archivo:

```
# nano -w /boot/grub/menu.lst
```

Y por último, eliminando los módulos instalados:

```
# rm -fr /lib/modules/2.6.17-gentoo-r4
```

10.3. Remover las fuentes de los kernels que ya no se usen

El mayor espacio que ocupa un kernel en el disco duro es en el directorio que contiene todo el código fuente. El directorio **/usr/src/linux-2.6.20-gentoo-r8** me consume **¡ 364 MB !**. Después del comando **make clean** el espacio ocupado descendió a **282 MB**.

Para eliminar un las fuentes de un kernel, por ejemplo el kernel 2.6.17 r4, ejecute:

```
rm -rf /usr/src/linux-2.6.17-gentoo-r4
```

11. Conclusiones

Compilar su propio kernel puede llegar a ser una de las cosas más fantásticas que haga con GNU/Linux o una de las más frustrantes. Pero hay algo en que cualquiera está de acuerdo: Los kernels más recientes soportan más hardware, tienen más *bugs* reparados y mejor desempeño. Por todo esto, vale la pena dedicarle tiempo y estudio a esta ardua labor.

Piense en esta gran virtud del Software Libre, usted lo puede estudiar y modificar. Tenga por seguro que al personalizar el kernel Linux está de verdad modificándolo a su gusto e incrementando las capacidades de su equipo.

En este documento se han escrito las experiencias en compilación del kernel Linux que ha vivido el autor. Le invito a que también forme parte de esta *aventura* y viva sus propias experiencias.

12. Referencias

Recomiendo ampliamente que lea los siguientes manuales.

Debian Kernel Handbook
`http://kernel-handbook.alioth.debian.org/`

initrd From Wikipedia, the free encyclopedia
`http://en.wikipedia.org/wiki/Initramfs`

Gentoo Handbook
`http://www.gentoo.org/doc/en/handbook/index.xml`

Guía de instalación de Debian GNU/Linux
`http://www.debian.org/releases/stable/i386/index.html.es`