

# Compilar un núcleo

## 1. Obtener las fuentes

### a. Fuentes oficiales

Las fuentes oficiales del núcleo se llaman vanilla. Un núcleo (o kernel) vanilla es un núcleo bruto, sin parches añadidos, procedente directamente de los desarrolladores que colaboran en el núcleo, y no ha sido adaptado aún a ninguna distribución.

Se obtienen las fuentes oficiales desde el sitio <https://www.kernel.org>. El comando **wget** lo hará por nosotros:

```
$ wget https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-5.5.2.tar.xz
```

Una vez se haga con las fuentes, debe configurar, compilar e instalar un núcleo, y si es preciso, crear un **initrd** antes de poder utilizarlo.

El núcleo se proporciona en forma de carpeta comprimida que debe abrir con las herramientas adecuadas.

El formato actual de compresión empleado es el formato XZ. Probablemente deberá instalar las herramientas xz-utils (Debian, Ubuntu) o xz (Fedora, Red Hat). La opción asociada de tar es "J".

```
# ls
*linux-5.5.2.tar.xz
# tar tvJf linux-5.5.2.tar.xz |more

drwxrwxr-x root/root    0 2020-02-04 19:18 linux-5.5.2/
-rw-rw-r-- root/root 15318 2020-02-04 19:18 linux-5.5.2/.clang-format
-rw-rw-r-- root/root   59 2020-02-04 19:18 linux-5.5.2/.cocciconfig
-rw-rw-r-- root/root   71 2020-02-04 19:18 linux-5.5.2/.get_maintainer.ignore
-rw-rw-r-- root/root   62 2020-02-04 19:18 linux-5.5.2/.gitattributes
-rw-rw-r-- root/root 1746 2020-02-04 19:18 linux-5.5.2/.gitignore
```

```

-rw-rw-r-- root/root    14461 2020-02-04 19:18 linux-5.5.2/.mailmap
-rw-rw-r-- root/root      423 2020-02-04 19:18 linux-5.5.2/COPYING
-rw-rw-r-- root/root   99600 2020-02-04 19:18 linux-5.5.2/CREDITS
drwxrwxr-x root/root      0 2020-02-04 19:18 linux-5.5.2/Documentation/
-rw-rw-r-- root/root     13 2020-02-04 19:18 linux-5.5.2/Documentation/.gitignore
drwxrwxr-x root/root      0 2020-02-04 19:18 linux-5.5.2/Documentation/ABI/
-rw-rw-r-- root/root    3821 2020-02-04 19:18 linux-5.5.2/Documentation/ABI/README
drwxrwxr-x root/root      0 2020-02-04 19:18 linux-5.5.2/Documentation/ABI/obsolete/
-rw-rw-r-- root/root    1296 2020-02-04 19:18 linux-5.5.2/Documentation/ABI/obsolete/
sysfs-bus-usb
-rw-rw-r-- root/root     883 2020-02-04 19:18 linux-5.5.2/Documentation/ABI/obsolete/
sysfs-class-dax
-rw-rw-r-- root/root    1402 2020-02-04 19:18 linux-5.5.2/Documentation/ABI/obsolete/
sysfs-class-net-batman-adv
-rw-rw-r-- root/root    4606 2020-02-04 19:18 linux-5.5.2/Documentation/ABI/obsolete/
sysfs-class-net-mesh
-rw-rw-r-- root/root     2023 2020-02-04 19:18 linux-5.5.2/Documentation/ABI/obsolete/
sysfs-class-typec

```

Se deben ubicar las fuentes del núcleo en `/usr/src/Linux`, `/usr/src/linux-sources-x.y.z` o `/usr/src/kernels`, según la distribución. Si en el sistema de archivos se encuentran varias versiones de los archivos fuente del núcleo, el archivo `Linux` puede ser un vínculo simbólico hacia la fuente del núcleo actual.

```

# ls -l
total 74944
lrwxrwxrwx 1 root src      13 2014-06-29 23:11 linux -> linux-3.16-rc3
drwxrwxr-x 23 root root 4096 2014-06-29 23:11 linux-3.16-rc3

```

## b. Fuentes de la distribución

Las distribuciones vienen con núcleos que menudo están parcheados. Estas modificaciones conllevan varios aspectos: drivers añadidos, backports (actualizaciones retroactivas, añadido de funcionalidades procedentes de una versión ulterior) de núcleos más recientes, actualizaciones de seguridad, funcionalidades añadidas, etc. Se aplican a un núcleo vanilla y a menudo en un orden preciso.

Compilar un núcleo no es una nadería. Si ha firmado un contrato de asistencia con el editor de la distribución (por ejemplo, para una versión Servidor de Red Hat), perderá dicha asistencia si compila el núcleo ested mismo. Si tiene problemas con el núcleo por defecto, debería pensar en:

- ˘ comprobar si una actualización oficial corrige su problema;
- ˘ subir su problema al foro de asistencia del editor.

Si a pesar de todo desea recompilar el núcleo, el trámite es el mismo que para las fuentes oficiales, sólo que debe instalar el paquete de las fuentes, lo que debería tener como efecto la instalación simultánea de todas las herramientas necesarias para la compilación.

El paquete se llama `kernel-sources` en OpenSUSE y Red Hat. En cuanto a este último, diríjase al sitio Red Hat, ya que el paquete **kernel-sources** no es necesario para compilar nuevos módulos (hace falta el paquete `kernel-devel`) y no es muy fácil obtenerlos. En Debian o Ubuntu, el paquete se llama **linux-source-3.x.y.z** (x, y y z representan la versión del núcleo) o **linux-source** (instale la última versión disponible). Debe instalar estos paquetes con las herramientas adaptadas a su distribución.

A continuación se verá la compilación del núcleo en una distribución Ubuntu 19.10, pero el método es el mismo para todas las demás distribuciones (a excepción de apt).

## 2. Las herramientas necesarias

Para compilar el núcleo de Linux, es necesario disponer de determinadas herramientas:

- ˘ el compilador C;
- ˘ las librerías de desarrollo C estándar;
- ˘ la librería `ncurses` (para `menuconfig`);
- ˘ la librería `qt3` o `qt4` (para `xconfig`);
- ˘ las herramientas `Make`;
- ˘ los `modutils`;
- ˘ `mkinitrd`;

- ~ concentración;
- ~ nervios de acero;
- ~ paciencia.

La distribución proporciona estas herramientas de forma automática, salvo las tres últimas...

He aquí las diferentes etapas que le permitirán instalar los paquetes necesarios para compilar el núcleo en una Ubuntu. Prevea 30 GB de espacio disponible en el disco para tener margen a la hora de la compilación donde descomprimirá las fuentes y se lanzará la construcción. Con respecto a la memoria, 1 GB (o menos) sería suficiente, prevea un poco más si está ejecutando un entorno gráfico. El autor dispone de 4 GB.

Para empezar, edite el archivo `/etc/apt/sources.list` y suprima los comentarios de las líneas siguientes:

```
deb-src http://fr.archive.ubuntu.com/ubuntu eoan main restricted
deb-src http://fr.archive.ubuntu.com/ubuntu eoan-updates main restricted
```

Actualice la base de datos de apt:

```
$ sudo apt-get update
```

Instale las dependencias que le permitirán construir el núcleo:

```
$ sudo apt-get build-dep linux linux-image-$(uname -r)
$ sudo apt-get install libncurses-dev flex bison openssl libssl-dev dkms
libelf-dev libudev-dev libpci-dev libiberty-dev autoconf
```

Descomprima las fuentes en el directorio actual, por ejemplo, en su directorio personal:

```
$ tar xJf linux-5.5.2.tar.xz
```

Entre en el directorio de las fuentes:

```
$ cd linux-5.5.2
```

## 3. Configuración

### a. El `.config`

Antes de lanzar la compilación del núcleo, debe seleccionar las opciones, funcionalidades y drivers que desea conservar o no. A continuación, le mostraremos varias formas de hacerlo. Luego se guarda esta configuración en un archivo `.config` de la carpeta de fuentes. Este archivo contiene un gran número de variables: cada una de ellas corresponde a una opción del núcleo y pueden adoptar tres valores:

- ~ **y**: la funcionalidad está presente e integrada dentro del núcleo monolítico, o, si depende de un módulo, integrada dentro de ese módulo;
- ~ **m**: se compilará la funcionalidad en forma de módulo;
- ~ **n**: la funcionalidad no está.

En el último caso, este valor está presente pocas veces. Basta con que la opción no esté en el archivo para que no se active durante la compilación. En este caso, se comenta la línea correspondiente simplemente con una `#` delante.

Si acaba de descomprimir un núcleo vanilla desde las fuentes, este archivo todavía no está presente, tendrá que generarlo o recuperarlo siguiendo las explicaciones de la sección siguiente.

El ejemplo siguiente muestra que la opción 64BIT (compilación para procesadores de 64 bits) está activa.

```
# grep -i 64bit .config
CONFIG_64BIT=y
```

El ejemplo siguiente muestra las líneas de configuración asociadas a las funcionalidades del sistema de archivos ext4. Se compila el soporte del sistema de archivos ext4 en forma

de módulo (primera línea). Las funcionalidades adicionales de ext4 (atributos extendidos, ACL, seguridad, atributos nfs4) dependen de la primera línea. Por lo tanto, estarán en el núcleo y no en uno o varios módulos aparte.

```
# grep -i ext4 .config
CONFIG_EXT4_FS=y
CONFIG_EXT4_USE_FOR_EXT23=y
CONFIG_EXT4_FS_POSIX_ACL=y
CONFIG_EXT4_FS_SECURITY=y
CONFIG_EXT4_DEBUG=y
```

Algunas opciones dependen de otras. A no ser que sepa con exactitud lo que hace, no debería editar el archivo `.config` manualmente para efectuar modificaciones en él, pues corre el riesgo de romper dependencias durante la compilación. Lo más sencillo consiste en pasar por las etapas de configuración expuestas a continuación.



Evite modificar `.config` manualmente. Utilice mejor las herramientas facilitadas desde las fuentes o por su distribución.

## b. Recuperar la configuración del núcleo

Puede acceder a la configuración actual del núcleo desde varios sitios. Si un núcleo (o sus fuentes) proceden de un paquete de la distribución, es probable que el archivo `.config` esté ya dentro de `/usr/src/linux` o en otro sitio. En este caso, debe consultar la documentación oficial.

El directorio `/boot` contiene con frecuencia una copia del `.config` que ha servido para la compilación del núcleo actual.

```
# ls -l /boot/config-*
/boot/config-5.3.0-26-generic
/boot/config-5.3.0-29-generic
```

En este caso, puede reutilizar esta configuración manualmente. Tóme la versión más cercana del núcleo que va a recompilar:

```
# cp /boot/config-5.3.0-29-generic .config
```

A menudo se configuran los núcleos con dos opciones interesantes, que frecuentemente están ausentes de muchas distribuciones, pero que se pueden considerar como uno fallo de seguridad:

```
# grep -i KCONF .config
CONFIG_IKCONFIG=y
CONFIG_IKCONFIG_PROC=y
```

La primera permite colocar el contenido del `.config` en el propio núcleo durante la compilación. La segunda permite acceder a esta configuración desde el sistema de archivos virtual `/proc` mediante el archivo `/proc/config.gz`. Se comprime el pseudoarchivo en formato gzip. Para leerlo, utilice el comando **zcat**.

```
# zcat /proc/config.gz > /usr/src/linux/.config
```

### c. make oldconfig

El método precedente presenta unos inconvenientes en particular cuando se trata de recuperar la configuración de un núcleo más reciente o más antiguo:

- ~ puede que algunas funcionalidades hayan desaparecido;
- ~ y que se hayan añadido otras.

En este caso, partir de un archivo de configuración inadecuado puede tener consecuencias nefastas. Para evitar cualquier problema, lo mejor es utilizar la posibilidad que ofrecen las fuentes del núcleo (o más bien Makefile): recuperar la antigua configuración, analizarla, indicar los cambios y preguntar qué hacer. Esto se hace con el comando **make oldconfig**.

En el ejemplo siguiente, se vuelve a coger la configuración de un núcleo 5.3.0 de origen Ubuntu 19.10 para un núcleo vanilla 5.5.2. Hemos truncado muchos de los avisos que aparecen, ya que el núcleo vanilla no contiene ciertas opciones procedentes de parches específicos del núcleo de la distribución. A continuación, el nuevo núcleo dispone de opciones que no estaban en origen. Para estas se muestra **NEW** y debe contestar a cada una de las preguntas, lo que resulta fastidioso.

```
CPU controller (CGROUP_SCHED) [Y/?] y
Group scheduling for SCHED_OTHER (FAIR_GROUP_SCHED) [Y/?] y
CPU bandwidth provisioning for FAIR_GROUP_SCHED (CFS_BANDWIDTH) [Y/n/?] y
Group scheduling for SCHED_RR/FIFO (RT_GROUP_SCHED) [N/y/?] n
Utilization clamping per group of tasks (UCLAMP_TASK_GROUP) [N/y/?] (NEW)
...
#
# configuration written to .config
#
```



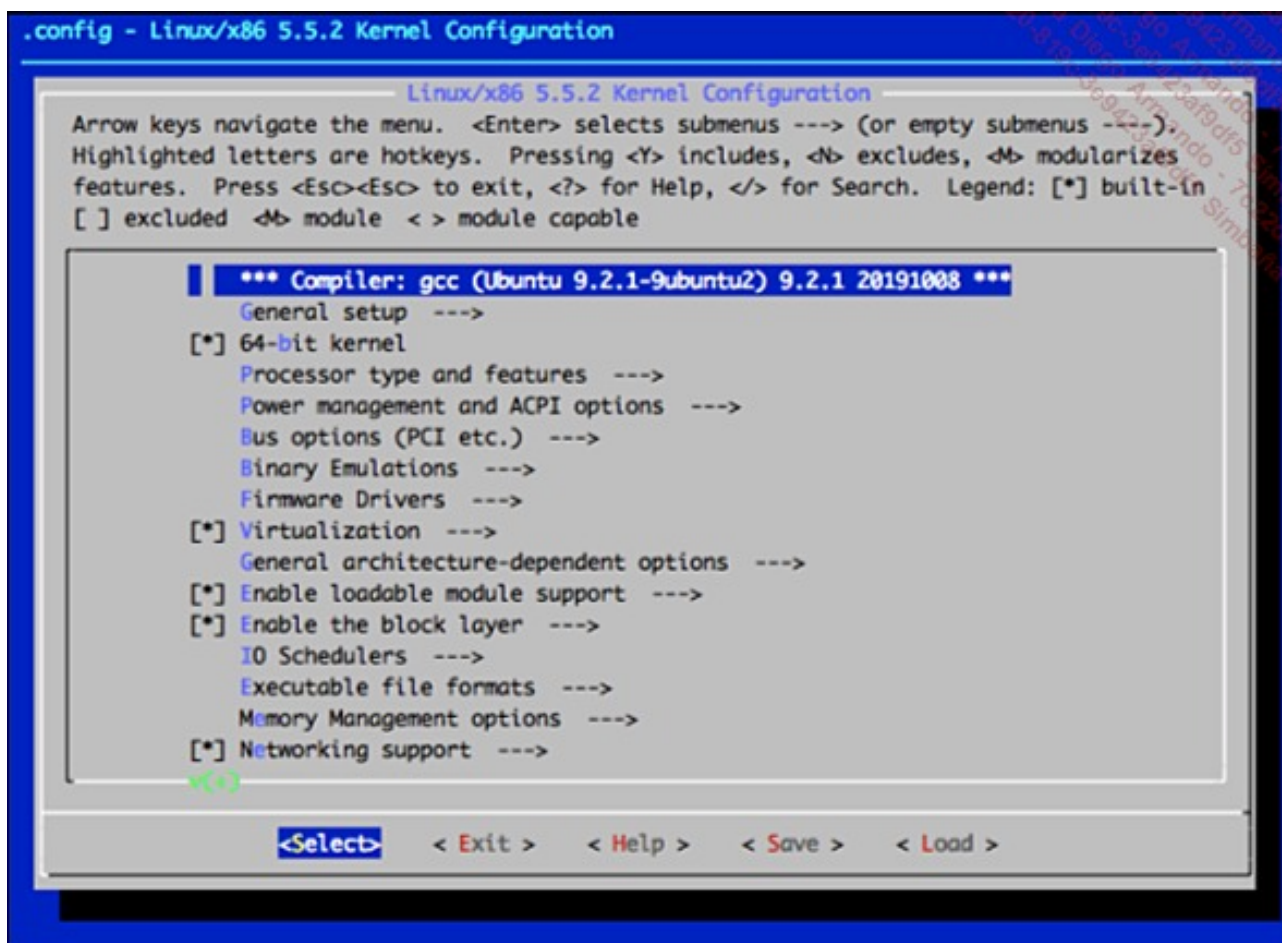
Si pulsa la tecla [Entrar], se activa la opción por defecto. Puede seguir pulsando la tecla [Entrar] hasta el final y luego modificar la configuración con la interfaz textual o gráfica.

#### d. make menuconfig

El método anterior es práctico cuando se trata de una migración. Sin embargo, quizás querrá utilizar algo más ameno. Si no dispone de una interfaz gráfica (a menudo ausente en los servidores, ya que es inútil), puede configurar su núcleo pasando por una interfaz en modo consola.

```
# make menuconfig
```





Las opciones de compilación del núcleo empleando menuconfig

Las instrucciones se incluyen en la interfaz (teclea [Entrar], [Y], [N], [M], [?], [/], [Esc]). Observe que si pulsa la tecla [Espacio] se desplegarán todas las opciones posibles. No dude en utilizar la tecla de ayuda para (intentar) entender para qué sirve una opción.

Una vez elegidas las opciones, salga y se generará el archivo .config de manera automática.

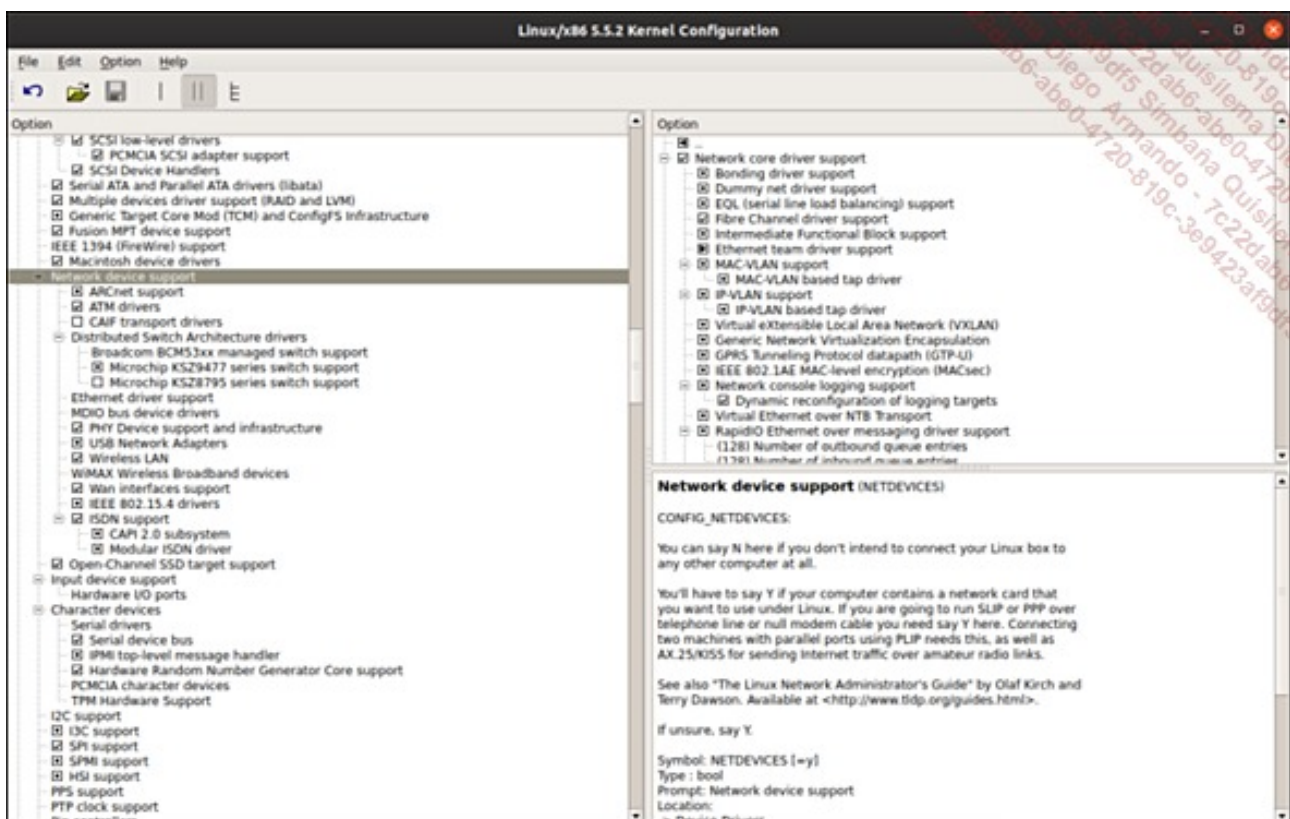
### e. make xconfig

Si dispone de un entorno gráfico, he aquí la mejor elección posible. Dispone de una interfaz gráfica para configurar su núcleo. La lista de entradas principales está a la izquierda, en forma de árbol. Elija a la derecha las opciones asociadas. En el cuadro inferior, obtendrá ayuda para cada una de las opciones. Los menús le permiten efectuar una búsqueda entre los valores.

Tendrá que instalar las librerías de desarrollo qt4 (generalmente qt4-devel, o qt-devel) al igual que el compilador c++ (gcc-c++).

Desde la interfaz gráfica, puede cargar una configuración procedente de otro archivo de configuración o exportarla hacia un archivo que elija.

```
# make xconfig
```



Las opciones de compilación empleando xconfig

Si ejecuta make config, la interfaz utilizada será gtk en vez de qt. Será necesario instalar las librerías de desarrollo gtk.

## f. Pistas para optimización

Las distribuciones Linux suministran núcleos genéricos. Algunas proponen núcleos específicos para algunos tipos de procesadores (Intel, AMD, etc.). Según las arquitecturas soportadas por las distribuciones, versiones de 32 bits y 64 bits. Estas permiten

aprovechar los procesadores recientes.

Estos núcleos se diseñan para funcionar en un máximo de máquinas, procesadores y dispositivos. Se compilan con opciones estándares y la casi totalidad del soporte de los diferentes periféricos. Así pues, no están optimizados para algunos usos. Le parecerá muy interesante lo que puede aportar la posibilidad de adaptar el núcleo a su máquina:

- ˘ Una adecuación perfecta de las optimizaciones del núcleo para su hardware.
- ˘ Mayor eficacia según el tipo de uso (servidor, estación de trabajo).
- ˘ El soporte de nuevos materiales (núcleo más reciente).
- ˘ Una reducción del volumen en disco ocupado por los módulos inútiles.



En cualquier caso, si compila un nuevo núcleo, no desinstale ni borre el antiguo. Consérvelo: en caso de problemas con su núcleo personalizado, tendrá la posibilidad de volver atrás.

Algunas pistas para mejorar su núcleo:

- ➔ Active las optimizaciones del núcleo para su procesador. En la entrada **Processor type and features** seleccione, por ejemplo, **Core2 / Newer Xeon** si su máquina dispone de un procesador Core 2 Duo o superior (i3, i5, i7).
- ➔ Active **Multicore Scheduler Support** y las opciones asociadas.
- ➔ Modifique el valor **Preemption Model** a **Preemptible kernel** o **Voluntary kernel preemption** y marque la casilla **Preempt the Big Kernel Clock** para reducir el efecto de latencia del núcleo (para establecer dicho modo en el proceso).
- ➔ Modifique el valor del **Timer Frequency** a 300 Hz o 1000 Hz para reducir el tiempo de respuesta a los eventos.
- ➔ En **CPU Frequency Scaling**, también puede aumentar ligeramente la velocidad de inicio del núcleo pasando el valor **Default CPUfreq Governor** a **performance**. Esto no cambia nada de momento. Puede seguir ajustando después la velocidad del procesador.

→ Suprima luego los diferentes módulos que no va a necesitar nunca. Por ejemplo, no hace falta conservar las tarjetas Token Ring o ISDN si no las utiliza jamás.

En cuanto a este último punto, tenga cuidado de no suprimir cualquier cosa. No olvide que, si cambia de máquina, a diferencia de Windows, no tiene que volver a instalar obligatoriamente Linux si el núcleo es bastante genérico (quizá tendrá que bootear en single para modificar los módulos cargados en el inicio). El autor migró de un Core 2 Duo a un Core i5 (con cambio de placa base, tarjeta gráfica, etc.) sin necesidad de reinstalar completamente su sistema. El nuevo sistema puede necesitar módulos desafortunadamente suprimidos.

Piense también en los soportes y periféricos en hotplug o en hotswap. Si por error ha suprimido el driver o una opción del driver, tendrá que recompilar el núcleo o reiniciar a partir del antiguo que haya conservado.



Una buena idea con los núcleos por defecto de las distribuciones consiste en tocar únicamente las opciones de optimización, como las que hemos visto antes, dejando todas las demás opciones por defecto.

### ¿32 o 64 bits?

Esta pregunta ya no está de actualidad. Casi todas las máquinas vendidas hoy en día, incluso los ordenadores portátiles, disponen de un conjunto de instrucciones de 64 bits. Hasta la llegada de Windows 7, los ordenadores disponibles para el gran público se vendían con sistemas de 32 bits (Windows XP o Vista, principalmente), a pesar de que existían las versiones de 64 bits. Los 64 bits ofrecen más ventajas que inconvenientes:

- ~ Mejor gestión de la memoria (adiós a la limitación de 4 GB).
- ~ Mayor eficacia en las aplicaciones optimizadas.
- ~ Compatibilidad en el futuro garantiza.
- ~ Adiós al efecto 2000 (consecuencia de pasar a cero todas las fechas y regresar al 13 de diciembre de 1901) ¡Unix en enero de 2038!

A modo de comparación, la transcodificación de un DVD en DVIX con dvdrip (opciones: de una pasada, desenlace inteligente, gran tamaño de redimensionamiento HQ, mp3 192 kbits) se ejecutaba a una velocidad de 27 a 30 imágenes por segundo en OpenSUSE 10.3 en 32 bits (Core 2 Duo e6750). En la misma máquina, la velocidad era de 40-42 imágenes por segundo (en una instalación optimizada para trabajar en 64 bits). O sea, una ganancia media del 25 al 30 %. Se constata lo mismo con un cálculo súper-pi optimizado. La ganancia puede ser real, incluso si no es significativa en un entorno ofimático.

Dado que se puede obtener el núcleo de Linux mediante archivos fuente y que la mayoría de los drivers están incluidos en las distribuciones o en forma de fuentes, cualquier hardware soportado en 32 bits por Linux lo estará también en 64 bits. Se facilitan algunos módulos propietarios (ati, nvidia) en 64 bits. Las distribuciones de 64 bits proporcionan lo necesario (las librerías) para hacer funcionar las aplicaciones de 32 bits de manera nativa.

Si tiene esta posibilidad, pruebe una versión de 64 bits de Linux.



Si vuelve a compilar un núcleo con las extensiones de 64 bits desde una distribución de 32 bits, únicamente el núcleo en 64 bits funcionará. Habría que instalar luego todas las librerías y todas las herramientas en 64 bits, lo que resulta complejo y largo. En este caso, lo mejor es volver a instalar el sistema.

## 4. Compilación

Ahora que se ha configurado el núcleo, puede iniciar la compilación. Esta etapa es la más larga. El núcleo contiene millones de líneas en lenguaje C y ensamblador. Según las opciones seleccionadas (y, sobre todo, los módulos incluidos), el nivel de ocupación de su máquina y el rendimiento de su procesador, esta etapa requiere desde unos minutos ¡hasta unas horas! En cualquier caso, tiene tiempo para hacer otra cosa.

Inicie el comando **make**. Si dispone de varios procesadores o de varios cores, agregue el parámetro **-j x**, donde x es el número de cores: la compilación se hará en paralelo y funcionará mucho más rápido.

```

# make -j 2
SYSTBL arch/x86/ include/generated/asm/syscalls_32.h
WRAP arch/x86/ include/generated/uapi/asm/ bpf_perf_event.h
WRAP arch/x86/include/generated/asm/errno.h
WRAP arch/x86/include/generated/uapi/asm/fcntl.h
WRAP arch/x86/include/generated/uapi/asm/ioctl.h
WRAP arch/x86/include/generated/uapi/asm/ioctls.h
WRAP arch/x86/include/generated/uapi/asm/ipcbuf.h
WRAP arch/x86/include/generated/uapi/asm/param.h
WRAP arch/x86/include/generated/uapi/asm/poll.h
WRAP arch/x86/include/generated/uapi/asm/resource.h
WRAP arch/x86/include/generated/uapi/asm/socket.h
WRAP arch/x86/include/generated/uapi/asm/sockios.h
WRAP arch/x86/include/generated/uapi/asm/termbits.h
WRAP arch/x86/include/generated/uapi/asm/termios.h
WRAP arch/x86/include/generated/uapi/asm/types.h
UPD include/generated/uapi/linux/version.h
HOSTCC arch/x86/tools/relocs_32.o
UPD include/config/kernel.release
...
BUILD arch/x86/boot/bzImage Setup is 18044 bytes (padded to 18432 bytes).
System is 11429 kB
CRC 5f220703
Kernel: arch/x86/boot/bzImage is ready (#1)

Building modules, stage 2.
...
CC [M] drivers/media/rc/keymaps/rc-ati-tv-wonder-hd-600.mod.o
CC [M] drivers/media/rc/keymaps/rc-ati-x10.mod.o
CC [M] drivers/media/rc/keymaps/rc-avermedia-a16d.mod.o
CC [M] drivers/media/rc/keymaps/rc-avermedia-cardbus.mod.o
CC [M] drivers/media/rc/keymaps/rc-avermedia-dvbt.mod.o
...
LD [M] sound/xen/snd_xen_front.ko
LD [M] virt/lib/irqbypass.ko

```

La primera columna representa la acción que se realiza en el archivo representado por la segunda columna. CC indica una compilación; LD, la edición de los vínculos; una [M], la acción sobre un módulo, IHEX un archivo binario, IHEX2FW un firmware, etc.

## 5. Instalación

Si la compilación se efectúa sin error, le quedan dos acciones por hacer. Primero instale los módulos. El comando siguiente los instala en `/lib/module/<version_núcleo>` y crea el archivo de las dependencias asociadas.

```
# make modules_install
INSTALL arch/x86/crypto/aegis128-aesni.ko
INSTALL arch/x86/crypto/aesni-intel.ko
INSTALL arch/x86/crypto/blowfish-x86_64.ko
INSTALL arch/x86/crypto/camellia-aesni-avx-x86_64.ko
INSTALL arch/x86/crypto/camellia-aesni-avx2.ko
...
INSTALL sound/x86/snd-hdmi-lpe-audio.ko
INSTALL sound/xen/snd_xen_front.ko
INSTALL virt/lib/irqbypass.ko
DEPMOD 5.5.2
```

Este segundo comando vuelve a copiar el núcleo y lo necesario asociado en `/boot`. Según las distribuciones, va a crear también el `initrd` asociado y posiblemente modificar la configuración del gestor de arranque GRUB o GRUB2.

```
# make install
sh ./arch/x86/boot/install.sh 5.5.2 arch/x86/boot/bzImage \
System.map "/boot"
run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 5.5.2
/boot/vmlinuz-5.5.2
run-parts: executing /etc/kernel/postinst.d/dkms 5.5.2 /boot/vmlinuz-5.5.2
* dkms: running auto installation service for kernel 5.5.2
Kernel preparation unnecessary for this kernel. Skipping...

Building module:
cleaning build area...
make -j2 KERNELRELEASE=5.5.2 -C /lib/modules/5.5.2/build M=/var/lib/dkms/virtualbox-guest/
6.0.14/build....(exit status: 0)
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 5.5.2
```

```

/boot/vmlinuz-5.5.2
update-initramfs: Generating /boot/initrd.img-5.5.2
I: The initramfs will attempt to resume from /dev/sda4
I: (UUID=25509077-22e0-476d-a9ea-87a85106ce96
I: Set the RESUME variable to override this.
run-parts: executing /etc/kernel/postinst.d/unattended-upgrades 5.5.2
/boot/vmlinuz-5.5.2
run-parts: executing /etc/kernel/postinst.d/update-notifier 5.5.2 /boot/vmlinuz-5.5.2
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 5.5.2
/boot/vmlinuz-5.5.2
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.5.2
Found initrd image: /boot/initrd.img-5.5.2
Found linux image: /boot/vmlinuz-5.3.0-29-generic
Found initrd image: /boot/initrd.img-5.3.0-29-generic
Found linux image: /boot/vmlinuz-5.3.0-26-generic
Found initrd image: /boot/initrd.img-5.3.0-26-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
done

```

El resultado en /boot es el siguiente:

```

# ls -l /boot|grep 5.5.2
config-5.5.2
initrd.img-5.5.2
System.map-5.5.2
vmlinuz-5.5.2

```

Observe que no se ha reconstruido el initrd en este caso, lo que no es normal y debería funcionar con la salida de un núcleo estable. Puede reconstruirlo usted mismo, por ejemplo en Debian o Ubuntu, como hemos visto anteriormente:

```

# cd /boot

```



```
# mkinitramfs -o initrd.img-5.5.2 5.5.2
```

Respete el orden precedente. Si instala el núcleo antes que sus módulos, no se podrá volver a construir el initrd, ya que aún no se han instalado los módulos que deberían estar presentes. Puede que no se genere el initrd por defecto. En este caso, tendrá que volver a ejecutar este comando después de la instalación con los parámetros correspondientes al núcleo instalado.

Si el nuevo núcleo no se tiene en cuenta, vuelva a ejecutar `grub2-mkconfig` o `grub-mkconfig`:

```
# grub-mkconfig -o /boot/grub/grub.cfg
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.5.2
Found initrd image: /boot/initrd.img-5.5.2
Found linux image: /boot/vmlinuz-5.3.0-29-generic
Found initrd image: /boot/initrd.img-5.3.0-29-generic
Found linux image: /boot/vmlinuz-5.3.0-26-generic
Found initrd image: /boot/initrd.img-5.3.0-26-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
done
```



No tiene por qué reinstalar GRUB o GRUB2 cuando modifica su archivo de configuración.

## 6. Test

Si todas las etapas anteriores han culminado con éxito, sólo le queda encender de nuevo

el ordenador y seleccionar el nuevo núcleo en el momento de la carga. Si el boot termina correctamente (acceso a la consola o entorno gráfico), abra una consola y verifique la versión del sistema.

```
$ uname -a  
Linux ubuntu 5.5.2 #1 SMP Wed Feb 5 19:40:41 CET 2020 x86_64 x86_64 x86_64 GNU/Linux
```

## 7. Otras opciones

Tras la compilación, los archivos intermediarios (archivos objetos) ocupan mucho espacio. Para suprimirlos, utilice el comando siguiente: `# make clean`

Si su distribución cuenta con el sistema rpm de paquetería, puede crear los paquetes del núcleo (fuentes, headers, núcleo) con el comando siguiente: `# make rpm`