

Construyendo el kernel de Linux usando git

Nota del autor: Estoy escribiendo esto de la manera más imparcial que puedo, porque creo que el usuario debería decidir qué es lo mejor para él.

Dónde construir

Hay dos lugares comunes para construir el núcleo. Es importante que dondequiera que compile el kernel, sea una ubicación que no cambie ni corra el riesgo de eliminación o corrupción. Por ejemplo, **NO** compile en /tmp .

1. Puede construir el kernel en /usr/src tendrá que ser root para construir el kernel. Esto tiene la ventaja de que funciona bien para sistemas multiusuario.

```
mkdir -p /usr/src
cd /usr/src
su
```

2. Puede compilar el kernel en su directorio de inicio en un directorio que no vaya a eliminar. Por ejemplo, puede usar ~/.local/src o algo similar. Puede construir el kernel como un usuario regular. La desventaja es que otros usuarios en un sistema multiusuario no tendrán acceso a la fuente.

```
mkdir -p ~/.local/src
cd ~/.local/src
```

Dónde obtener la fuente del kernel

Hay dos formas de obtener la fuente del kernel. Asumiremos que la versión es la siguiente.

```
version=3.10.24
base=3.10
```

1. Puede usar git para obtener la fuente del kernel. Esto tiene la ventaja de que evitará problemas si [linux.org está roto](https://lwn.net/Articles/lwn2010-05-19). Una desventaja es que el directorio de origen puede aumentar de tamaño a medida que se extraen nuevas versiones.

```
git clone --depth 1
git://git.kernel.org/pub/scm/linux/kernel/git/stable/linux-stable.git # Esto
solo debe ejecutarse una vez al principio
cd linux-stable
git pull # ejecute esto cada vez para actualizar la fuente
git checkout v$version # DEBE verificar una versión del kernel o obtendrá la
versión base, es decir, 3.10
git log # simplemente verifique para asegurarse de que la parte superior del
registro indique la versión que desea
```

git tag -v v\$version # esto no es necesario, pero valida la fuente usando una clave pública de gpg, que necesita importar

Para importar fácilmente una clave gpg, necesita la ID de clave RSA 0000000E. Ejecute

```
gpg --search 0000000E
```



y elija la clave que desea importar, o si eso no funciona, ejecute

```
gpg --keyserver wwwkeys.pgp.net --recv-keys 0000000E
```

La clave aquí es solo un ejemplo.

2. Puede descargar el código fuente de [<http://kernel.org/>]. La mejor manera de mantenerse al día es descargar el archivo comprimido de la versión base y luego descargar el parche de versión individual que desee, y aplicarlo antes de compilar.

```
kernelstie="ftp://www.kernel.org/pub/linux/kernel/v3.x"
lftp -c "open $kernelstie; get linux-$base.tar.xz linux-$base.tar.sign"
unxz -k "linux-$base.tar.xz"
gpg --verify "linux-$base.tar.sign"
# Lo siguiente se hace con cada nueva actualización del kernel
tar -xf "linux-$base.tar.xz"
mv "linux-$base" "linux-$version"
cd "linux-$version"
lftp -c "open $kernelstie; get patch-$version.xz patch-$version.sign"
unxz "patch-$version.xz"
gpg --verify "patch-$version.sign"
patch -p1 < "patch-$version"
```



La aplicación de parches a un kernel puede ser confusa, porque los parches son relativos a la versión básica del kernel. Digamos que actualmente tiene la fuente del kernel para 3.10.24, y desea actualizar a 3.10.25. Puede revertir el parche 3.10.24 (regresándolo a 3.10), o puede descargar la versión 3.10 del kernel y aplicar el parche 3.10.25. Si intenta aplicar el parche 3.10.25 sobre 3.10.24, fallará. Por lo tanto, se recomienda mantener el código fuente para la versión base en el directorio de compilación del kernel, especialmente si planea actualizar regularmente su kernel para obtener soluciones de seguridad.

Aquí hay un script que descarga el último parche para el kernel en ejecución actual. Se supone que ya tiene un tarball comprimido xz en el directorio de compilación del kernel.

[kernupd](#)

```
#!/bin/sh

error () # error
{
    echo "$1"
    exit 1
}

kerneldir="$HOME/.local/src"
kernelstite="ftp://www.kernel.org/pub/linux/kernel/v3.x"
currentver="$(uname -r)"
basever="$(uname -r | rev | cut -d. -f1 --complement | rev)"
newver="$(lftp -c "open $kernelstite; ls" | awk '{ print $9 }' | grep
"patch-$basever.*\.xz" | sort -V | tail -n1 | sed 's/patch-//' | sed
's/\.xz//')"

if test "$currentver" = "$newver"
then
    echo 'Kernel is up to date.'
else
    echo "Updating kernel to $newver"
    cd "$kerneldir"
    tar -xf "linux-$basever.tar.xz" || exit 1
    mv "linux-$basever" "linux-$newver" || exit 1
    cd "linux-$newver"
    lftp -c "open $kernelstite; get patch-$newver.xz patch-$newver.sign"
    || exit 1
    unxz "patch-$newver.xz"
    gpg --verify "patch-$newver.sign" || exit 1
    patch -p1 < "patch-$newver" || exit 1
    make mrproper
    zcat /proc/config.gz > .config
    make oldconfig
    echo "cd $PWD"
fi

exit 0
```

Cómo construir

Primero ejecute lo siguiente para asegurarse de que todo esté configurado correctamente para una compilación.

```
make clean
make mrproper # tenga en cuenta que esto elimina cualquier archivo .config
en el directorio actual
```

Tiene muchas opciones cuando se trata de cómo construir el Kernel. Es mejor leer el archivo README

en el directorio de la fuente del kernel para obtener todas las opciones actuales, así como también `Documentation /kbuild/kconfig.txt` para obtener detalles sobre todas las opciones.

README

- Alternative configuration commands are:

```
"make config"      Plain text interface.

"make menuconfig"  Text based color menus, radiolists & dialogs.

"make nconfig"     Enhanced text based color menus.

"make xconfig"     X windows (Qt) based configuration tool.

"make gconfig"     X windows (Gtk) based configuration tool.

"make oldconfig"   Default all questions based on the contents of
                  your existing ./config file and asking about
                  new config symbols.

"make silentoldconfig"
                  Like above, but avoids cluttering the screen
                  with questions already answered.
                  Additionally updates the dependencies.

"make olddefconfig"
                  Like above, but sets new symbols to their
default          values without prompting.

"make defconfig"   Create a ./config file by using the default
                  symbol values from either arch/$ARCH/defconfig
                  or arch/$ARCH/configs/${PLATFORM}_defconfig,
                  depending on the architecture.

"make ${PLATFORM}_defconfig"
                  Create a ./config file by using the default
                  symbol values from
                  arch/$ARCH/configs/${PLATFORM}_defconfig.
                  Use "make help" to get a list of all available
                  platforms of your architecture.

"make allyesconfig"
                  Create a ./config file by setting symbol
                  values to 'y' as much as possible.

"make allmodconfig"
                  Create a ./config file by setting symbol
```

```

values to 'm' as much as possible.

"make allnoconfig" Create a ./config file by setting symbol
values to 'n' as much as possible.

"make randconfig" Create a ./config file by setting symbol
values to random values.

"make localmodconfig" Create a config based on current config and
loaded modules (lsmod). Disables any module
option that is not needed for the loaded
modules.

To create a localmodconfig for another
machine,

store the lsmod of that machine into a file
and pass it in as a LSMOD parameter.

target$ lsmod > /tmp/mylsmod
target$ scp /tmp/mylsmod host:/tmp

host$ make LSMOD=/tmp/mylsmod localmodconfig

The above also works when cross compiling.

"make localyesconfig" Similar to localmodconfig, except it will
convert

all module options to built in (=y) options.

You can find more information on using the Linux kernel config tools
in Documentation/kbuild/kconfig.txt.

```

Otra opción más es comenzar con una de las configuraciones de kernel de slackware que se encuentran en el directorio `source /k` de sus medios de instalación de slackware. Simplemente copie el archivo de configuración, cámbiele el nombre a `.config` y ejecute uno de los comandos de configuración, normalmente `make oldconfig` si está compilando una versión del kernel más nueva que el archivo de configuración en el que se copió.

También puede usar la configuración de kernel de su kernel actualmente en ejecución ubicado en `/proc/config.gz` ejecutando

```
zcat /proc/config.gz > .config
```

Sin embargo, tenga en cuenta que `/proc/config.gz` puede no existir si el kernel no está configurado para exponerlo.



Si elige compilar en `/usr/src` y usa `xconfig` o `gconfig`, lea [howtos: slackware_admin: kernelbuilding # x_and_su](#)

Qué configurar



Primero, asegúrese de entender la diferencia entre el **módulo integrado**(Built-in) y el **módulo**. **Integrado** significa que el controlador o la característica se incluye en el kernel y el cargador de arranque lo carga en la RAM. **Módulo** significa que el controlador está construido como un módulo y es cargado por el núcleo una vez que monta el sistema de archivos. Esto es muy importante, ya que para cargar un módulo desde el sistema de archivos, el kernel necesita controladores para manejar el controlador IDE o SATA, así como el sistema de archivos en el HDD o SDD. **Debe** integrar los módulos de controlador IDE/SATA (PATA/SATA/AHCI) necesarios para acceder a su HDD/SDD, así como el controlador de sistema de archivos necesario para montar el sistema de archivos en el que se encuentran los módulos **o** puede [crear un initrd](#). Si no lo hace, se producirá el error "Kernel Panic-not syncing: VFS: unable to mount root fs on unknown block(8,2)" (no se puede montar fs raíz en un bloque desconocido).



Por lo tanto, si construye algo como un módulo, ¿por qué no hacer que todo sea Built-in? Ciertamente, esta es una opción, y esto es lo que se hace para el enorme kernel slackware, porque no tiene acceso a los módulos. Sin embargo, en una máquina antigua notará una diferencia entre el arranque del gran kernel (lento) y el arranque del kernel genérico (rápido). Esto se debe a que cargar un kernel grande en RAM lleva más tiempo que uno más pequeño. Hay otro problema con la construcción de todo en el kernel, ya que algunos controladores pueden entrar en conflicto entre sí y no podrá descargar los módulos porque están integrados. Por lo general, puede evitar que los controladores incorporados funcionen en el arranque utilizando la línea de comando del kernel que puede configurar en la configuración de su cargador de arranque, por ejemplo, `/etc/lilo.conf` .

Aspectos destacados del kernel config

Lo que sigue es una documentación de algunas de las características más importantes o interesantes del kernel. Esto no será completo, porque hay toneladas y toneladas de opciones para configurar. Solo se destacarán las opciones consideradas más interesantes o importantes que el resto. Lea la Ayuda en cada opción para más información. Se presentan en el mismo orden en que están en la configuración del kernel junto con dónde encontrarlos.

General setup

Local version - append to kernel release

Esta opción le permite agregar una cadena al final de la versión del kernel, en caso de que quiera instalar más de un kernel de la misma versión.

Kernel compression mode

Esto le permite seleccionar el modo de compresión del kernel. LZMA es la buena opción por defecto. La compresión más alta (LZMA / XZ) requiere más RAM y el uso del procesador, pero es más rápido de cargar desde el disco a la RAM. La compresión más baja (Gzip / LZO) requiere menos uso de RAM y procesador, pero es más lento de cargar desde el disco a la RAM.

- Kernel .config support

- Enable access to .config through /proc/config.gz

Asegúrese de habilitar esto para que pueda acceder fácilmente a la configuración actual de su kernel en /proc/config.gz

- Automatic process group scheduling

Esta opción puede mejorar en gran medida el rendimiento de la capacidad de respuesta de las máquinas multihilo. Por lo tanto, puede ejecutar `make -j4` sin hacer que otros programas tartamudeen.

- Initial RAM filesystem and RAM disk (initramfs/initrd) support

Asegúrese de habilitar esto si va a hacer un initrd.

- Disable heap randomization

Esta opción debería estar deshabilitada en máquinas fabricadas después del año 2000 por razones de seguridad.

- Optimize very unlikely/likely branches

Puede aumentar el rendimiento del núcleo.

Enable the block layer

- Partition Types —>

- EFI GUID Partition support

Habilite esto si planea usar el nuevo esquema de partición GPT y desea iniciar desde UEFI.

- IO Schedulers —>

- Deadline I/O scheduler

Causas mejora de rendimiento si utiliza el sistema de archivos JFS.

- CFQ I/O scheduler

- CFQ Group Scheduling support

Esta opción permite a CFQ reconocer grupos de tareas y controlar la asignación de ancho de banda del disco, y puede mejorar el rendimiento.

Puede habilitar ambos y probar cada uno ejecutando



```
cat /sys/block/sd?/queue/scheduler
echo deadline > /sys/block/sd?/queue/scheduler
echo cfq > /sys/block/sd?/queue/scheduler
```

Processor type and features

Symmetric multi-processing support

Esto debería estar habilitado para máquinas multi-core y multi-procesador.

Intel Low Power Subsystem Support

Habíltelo si tiene un sistema más nuevo con Intel Lynxpoint PCH. Compruebe la salida de `lspci` para Lynx Point .

Processor family

Para obtener el máximo rendimiento, debe elegir la familia de procesadores adecuada.

Para identificar un procesador desconocido, primero ejecute



```
cat /proc/cpuinfo
```

Si aún no puede saber qué familia de procesadores elegir, busque la “familia de cpu” y el “modelo” en línea o en la [gentoo wiki](#).

Maximum number of CPUs

Establece el número de CPU que tiene.

SMT (Hyperthreading) scheduler support

Esto es para máquinas Hyperthreading, es decir, núcleos virtuales.

Multi-core scheduler support

Esto es para máquinas multi-core, es decir, núcleos reales.



Puede habilitar ambos para estar seguro, en caso de que no sepa.

Preemption Model

Su selección depende de cómo use su computadora y hará una diferencia en cuanto al rendimiento.

No Forced Preemption (Server)

Alta latencia, rendimiento máximo. Bueno para servidores de datos.

Voluntary Kernel Preemption (Desktop)

Latencia media, rendimiento medio. Las aplicaciones seguirán funcionando sin problemas cuando el sistema esté bajo carga.

Preemptible Kernel (Low-Latency Desktop)

Baja latencia, bajo rendimiento. Bueno para servidores de medios, o sistemas embebidos de baja latencia.

Reroute for broken boot IRQs

Habilitar si tiene problemas con interrupciones espurias.

Machine Check / overheating reporting

Asegúrese de habilitar esto junto con Intel o AMD. Esto permite que el kernel responda al sobrecalentamiento del sistema.

Low address space to protect from user allocation

65536 para la mayoría de las arquitecturas, 32768 para ARM. Esto puede ayudar con errores de deferencia de puntero NULL.

Transparent Hugepage Support

Esto puede aumentar el rendimiento de las aplicaciones que requieren mucha memoria RAM a la vez, como p7zip, LZMA, LZMA2 (xz).

Transparent Hugepage Support sysfs defaults

Seleccionar siempre permitirá que funcionen las páginas de mensajes sin necesidad de montarlas o configurar la aplicación para usarlas, a diferencia de madvise.

Enable cleancache driver

Puede aumentar el rendimiento al reducir la E/S del disco. Sin embargo, no sé qué tan estable es.

MTRR cleanup support

Puede mejorar el rendimiento.

MTRR cleanup enable value (0-1)

1 para habilitarlo

MTRR cleanup spare reg num

Este es el número de registros MTRR inhabilitados/no utilizados/de repuesto.

Como ejemplo, ejecute `dmesg` y busque



```
MTRR variable ranges enabled:
 0 base 000000000 mask F80000000 write-back
 1 base 07EF00000 mask FFFF00000 uncachable
 2 base 07F000000 mask FFF000000 uncachable
 3 disabled
 4 disabled
 5 disabled
 6 disabled
```

Aquí, los registros 3,4,5,6 están deshabilitados/no utilizados /de repuesto, por lo que la limpieza de MTRR no registra el número total de 4 registros no utilizados.

EFI runtime service support

Habilítelo si desea arrancar desde UEFI.

EFI stub support

Habilítelo si desea arrancar desde UEFI.

Enable `-fstack-protector` buffer overflow detection

Puede evitar los desbordamientos de búfer en sistemas con gcc versión 4.2 y superiores.

Timer frequency

Siempre elija 1000Hz para los sistemas que necesitan ejecutar multimedia. Este número es proporcional a la capacidad de respuesta interactiva. Desea frecuencias más bajas en los servidores y frecuencias más altas en los escritorios. Sin embargo, si tiene Tickless System habilitado, las interrupciones del temporizador solo se activarán según sea necesario, por lo que puede ser mejor redondear.

Power management and ACPI options

ACPI

Habilitar siempre: Botón, Ventilador, Procesador, Zona térmica. Sin estos, su computadora (especialmente las computadoras portátiles) puede sobrecalentarse porque ACPI no puede acceder a la monitorización térmica ni a los ventiladores.

CPU Frequency scaling

CPU Frequency scaling

Habilítelo si tiene habilitado el escalado de frecuencia en UEFI / BIOS (Speedstep o similar).

'performance' governor

Esto ajusta la frecuencia de la CPU al máximo disponible.

'powersave' governor

Esto establece la frecuencia de la CPU al mínimo disponible.

'userspace' governor for userspace frequency scaling

Esto permite que los programas del espacio de usuario establezcan la frecuencia de la CPU.

'ondemand' cpufreq policy governor

Este regulador es recomendado para escritorios.



'conservative' cpufreq governor

Este regulador es recomendado para laptops / netbooks. Aunque es similar al gobernador 'ondemand', la frecuencia aumenta y disminuye gradualmente en lugar de saltar al 100% cuando se requiere velocidad.



x86 CPU frequency scaling drivers

Intel P state control

Este driver se excluye mutuamente con el controlador ACPI Processor P-States. Es un driver más nuevo para los procesadores Sandy Bridge y [puede causar problemas](#).



Processor Clocking Control interface driver

Esto solo es necesario para los servidores HP ProLiant, que utilizan esta interfaz. De lo contrario, desactívelo.



ACPI Processor P-States driver

Este es el driver recomendado para las CPU más nuevas Intel (mejorado) Speedstep habilitado y AMD K10 y más reciente.



AMD Opteron/Athlon64 PowerNow!

Esto es para los procesadores K8/early Opteron/Athlon 64.



Intel Enhanced SpeedStep (deprecated)

Esta es una opción en desuso que ha sido reemplazada por el controlador ACPI Processor P-States, así que deje esta opción deshabilitada.



Intel Pentium 4 clock modulation

Este es un truco para Pentium 4s que puede causar graves ralentizaciones y una latencia notable, así

que desactívelo.

Bus options

- Message Signaled Interrupts (MSI and MSI-X)

Esto puede descargar interrupciones IRQ utilizando MSI en su lugar. Sin embargo, si su BIOS tiene errores, es posible que deba desactivarse.

Executable file formats / Emulations

- Kernel support for scripts starting with #!

Usted **DEBE** decir sí aquí o no podrá ejecutar el script que comienza con #! Y luego el intérprete.

- IA32 Emulation

Permite la emulación de 32 bits a través de multi-lib en sistemas de 64 bits.

Device Drivers

- Block devices

- Normal floppy disk support

Debe ser un módulo o la unidad de disquete puede funcionar mal.

- Loopback device support

Requerido por algunos métodos de cifrado de disco.

- SCSI device support

- Asynchronous SCSI scanning

Se acelerará el arranque, pero todos los controladores SCSI **deben** estar integrados para que funcione correctamente.

- Serial ATA and Parallel ATA drivers

- AHCI SATA support

Si habilita AHCI en el BIOS, este **debe** estar integrado o ser parte de initrd. Si no usa AHCI, haga lo mismo pero para su controlador SATA o PATA.

- Multiple devices driver support (RAID and LVM)

- Device mapper support

- Crypt target support

Requerido por cryptsetup.

```
| | Input device support -->
| | | Keyboards -->
| | | | AT keyboard
```

Habilítelo si tiene un teclado estándar AT o PS / 2.

```
| | | Mice -->
| | | | PS/2 mouse
```

Habilítelo si tiene un ratón PS / 2.

```
| | Multimedia support -->
| | | Media USB Adapters -->
| | | | USB Video Class (UVC)
| | | | UVC input events device support
```

Si tiene una cámara web nueva, debe habilitarla o no funcionará.

```
| | Graphics support
| | | /dev/agpgart (AGP Support)
| | | Intel 440LX/BX/GX, I8xx and E7x05 chipset support
```

Habilítelo si tiene una tarjeta Intel integrada.

```
| | | Direct Rendering Manager
```

Esto casi siempre debería estar habilitado.

```
| | | Nouveau (nVidia) cards
```

Si usa Nouveau, solo necesita este controlador, ya que tiene soporte para framebuffer.

```
| | | Support for frame buffer devices
| | | | VESA VGA graphics support
```

Esto casi siempre debería estar habilitado.

 EFI-based Framebuffer Support

Debe habilitar esto si desea arrancar desde UEFI.

 nVidia Framebuffer Support

De nuevo, esto no es **NO** necesario para el nouveau.

 Sound card support Advanced Linux Sound Architecture PCI sound devices Intel HD Audio Pre-allocated buffer size for HD-audio driver

Se prefiere un búfer más grande (por ejemplo, 2048) para los sistemas que usan PulseAudio.

 USB support xHCI HCD (USB 3.0) support

Debe habilitar esto para los sistemas más nuevos.

 EHCI HCD (USB 2.0) support Improved Transaction Translator scheduling

Puede mejorar en gran medida las velocidades de transferencia a través de USB 2.0 en algunos casos.

 Real Time Clock PC-style 'CMOS'

Habilite esto o no podrá configurar el reloj desde Linux. A menudo se omite accidentalmente.

 Staging drivers —>

Evite los controladores en esta sección si valora la estabilidad del sistema.

 Generic Dynamic Voltage and Frequency Scaling (DVFS) support

Esto proporciona soporte de escala de frecuencia para dispositivos. También tiene gobernadores

similares a los de escalado de frecuencia de CPU. No estoy seguro de qué dispositivos requieren esto.

Firmware Drivers

- EFI (Extensible Firmware Interface) Support —>
 - EFI Variable Support via sysfs

Esta opción está en desuso a favor del sistema de archivos Variable EFI. Puede [causar problemas de inconsistencia de datos](#)

File systems

Usted **DEBE** bien incorporar el controlador para el sistema de archivos en el que se encuentran los módulos del kernel **O** bien crear un initrd, o el kernel **NO** se iniciará. También debe incorporar el controlador para el sistema de archivos en el que reside el núcleo en caso de que resida en un tipo de sistema de archivos diferente al de los módulos, por ejemplo. Partición EFI VFAT.

- DOS/FAT/NT Filesystems
 - MSDOS fs support
 - VFAT (Windows-95) fs support

Necesita estos incorporados si usa EFI. El núcleo arrancará si son módulos, pero las situaciones de tipo catch22 pueden resultar fácilmente.

- NTFS file system support
- NTFS write support

Necesita esto si desea escribir en sistemas de archivos NTFS.

- CD-ROM/DVD Filesystems
 - UDF file system support

Necesitará esto si planea leer o escribir discos con el sistema de archivos UDF.

- Miscellaneous filesystems —>
 - EFI Variable filesystem

Debe habilitar esto si desea arrancar desde UEFI.

Kernel hacking

- Magic SysRq key

Habilítelo si desea usar SysRq REISUB (p.cogx en dvorak) para apagar de forma segura un sistema

bloqueado.

Allow gcc to uninline functions marked 'inline'

Habilitar para gcc 4.x pero no para gcc 3.x

Security Options

Restrict unprivileged access to the kernel syslog

Desactive esto si desea que los usuarios puedan usar el comando 'dmesg'.

Cryptographic API

Asegúrese de incorporar todos los algoritmos que planea usar para la criptografía usando cryptsetup, especialmente si planea el cifrado completo del disco, de lo contrario no podrá descifrar su disco y, por lo tanto, no podrá iniciar. Tenga en cuenta que hay versiones optimizadas y de 64 bits para elegir.

Parallel crypto engine

Convierte un algoritmo criptográfico arbitrario en un algoritmo paralelo que se ejecuta en subprocesos del núcleo. Permite múltiples subprocesos de cualquier algoritmo criptográfico.

Library routines

JEDEC DDR data

Probablemente debería habilitar esto para que los datos JEDEC de su RAM estén disponibles para los controladores que lo necesitan.

Si se está preguntando qué drivers necesita, asegúrese de echar un vistazo a la salida de estos comandos.



```
lsmod  
/sbin/lspci -k  
lsusb  
dmesg
```

Es posible que desee considerar la desactivación de los drivers por etapas y los drivers EXPERIMENTALES y OBSOLETO si desea un núcleo estable y moderno. Algunos se pueden dejar por una buena razón.

Compilando

Para acelerar la compilación, puede usar la opción `-j` de `make`. El máximo al que se puede establecer es el número de núcleos o procesadores más uno. Sin embargo, si desea hacer algo

mientras el kernel se está construyendo, como navegar por la web, puede usar solo la cantidad de núcleos o procesadores.

```
cores=4
```

```
make -j$cores
```

Instalando



Antes de instalar el kernel, es posible que desee desinstalar los paquetes de Slackware que contienen los kernels, los módulos y la fuente predeterminados. También puede optar por dejar estos paquetes instalados si planea usar estos Kernels predeterminados. Haga lo que haga, **no** desinstale los encabezados del Kernel anterior. Consulte [guía de Alien Bob](#) para obtener más información sobre estos temas.

Primero, debe asegurarse de eliminar cualquier módulo de kernel previamente instalado en `/lib/modules/$version`.

```
su
rm -r /lib/modules/$version # Asegúrese de saber lo que se está eliminando
```

Ahora, puede instalar los nuevos módulos.

```
make modules_install
```

A continuación debería instalar el propio kernel. Esta secuencia de comandos instalará el kernel y se asegurará de que esté instalado correctamente.

[installkernel.sh](#)

```
#!/bin/sh
# installs kernel only, this should be run only from the kernel source
directory

error() # error
{
    echo "ERROR: $1"
    exit 1
}

# make sure we are root
if test ~ != '/root'
then
    error 'This script must be run as root'
fi
```

```
# remove the old
rm -f /boot/config.old
rm -f /boot/System.map.old
rm -f /boot/vmlinuz.old

# rename the present
mv /boot/config /boot/config.old
mv /boot/System.map /boot/System.map.old
mv /boot/vmlinuz /boot/vmlinuz.old

# copy in the new
cp arch/x86/boot/bzImage /boot/vmlinuz
cp System.map /boot
cp .config /boot/config

# for elilo
bootdir="/boot/efi/EFI/Slackware"
if test -d "$bootdir"
then
    cp arch/x86/boot/bzImage "$bootdir"/vmlinuz
fi

# change permissions of vmlinuz
chmod a-rwx,u+r /boot/vmlinuz

# check install
echo
if cmp arch/x86/boot/bzImage /boot/vmlinuz
then
    echo 'Kernel installed correctly'
else
    error 'Kernel install failed'
fi
if cmp System.map /boot/System.map
then
    echo 'System.map installed correctly'
else
    error 'System.map install failed'
fi
if cmp .config /boot/config
then
    echo 'Kernel config installed correctly'
else
    error 'Kernel config install failed'
fi
if test -d "$bootdir"
then
    if cmp arch/x86/boot/bzImage "$bootdir"/vmlinuz
    then
        echo 'kernel installed to EFI correctly'
    else
```

```
error 'kernel install to EFI failed'
fi
fi
echo
echo 'Kernel install completed successfully'
echo 'Remember to run lilo if you use it'
echo
exit 0
```

lilo



Si elige no incorporar los módulos de controlador IDE/SATA (PATA/SATA/AHCI) necesarios para acceder a su HDD/SDD, así como el controlador del sistema de archivos necesario para montar el sistema de archivos en el que se encuentran los módulos, usted **DEBE** [crear un initrd](#).

Si usa lilo, que es el gestor de arranque predeterminado en Slackware, debe editar `/etc/lilo.conf` y luego ejecutar

lilo

Aquí hay un ejemplo de `lilo.conf`



```
# append options here if you need any kernel parameters on boot
append=" vt.default_utf8=0"
# this should point to the device you want to boot
boot = /dev/sda
# not necessary, but lilo complains if it is not here
lba32
# the compact option speeds up boot time significantly, but may
not work on all systems
compact
# This is needed if you want lilo to prompt you for what to boot
prompt
# This is given in tenths of a second, so 600 for every minute.
You can comment this out if you single-boot.
timeout = 1200
# Override dangerous defaults that rewrite the partition table:
change-rules
    reset
# Normal VGA console, the safest choice.
# You can choose others if you want a framebuffer console, but
you must have framebuffer support or the screen will go black.
```



```
vga = normal
# End LILO global section
# Linux bootable partition config begins
# the path to the Linux kernel boot image
image = /boot/vmlinuz
# the partition where the Linux kernel is located
root = /dev/sda1
# how this entry is shown on the boot screen
label = Linux
read-only
# Linux bootable partition config ends
```

Fuentes

- El howto fue escrito por [H_TeXMeX_H](#)
- README es un extracto del README incluido con la fuente del núcleo.
- El script installkernel.sh fue escrito por [H_TeXMeX_H](#)
- El extracto de lilo.conf es del lilo.conf predeterminado que viene con Slackware más algunas opciones.
- Gracias a [Alien Bob](#) por su [kernel building howto](#). Me inspiró a escribir una guía de construcción de kernel reestructurada y actualizada.
- Actualizado por [metaschima](#)
- Traducido por: [Victor](#) 2019/02/04 21:07 (UTC)

[howtos](#), [author htexmexh](#), [kernel](#), [software](#)

From:
<https://docs.slackware.com/> - **SlackDocs**

Permanent link:
https://docs.slackware.com/es/howtos:slackware_admin:building_the_linux_kernel_using_git_repository

Last update: **2019/04/05 23:37 (UTC)**

