2.7- COMPILA EL KERNEL DE LINUX A TU MEDIDA (LINUX)

Cristóbal Suárez Abad

ADMINISTRACIÓN DE SISTEMAS OPERATIVOS - 2º ASIR

Índice

1. Instala el paquete correspondiente al núcleo que estás utilizando (sudo apt install linux-source) o puedes descargarlo (https://www.kernel.org)

sudo apt update && sudo apt upgrade -y

sudo apt install build-essential bc bison flex libssl-dev libncurses-dev libelf-dev dwarves fakeroot ccache rsync git wget curl

sudo apt install linux-source

o

sudo apt install linux-source linux-headers-$(uname -r)

NOTA: El código fuente se instala en **/usr/src/linux-source-<versión>.tar.xz.**

2. Crea un directorio para trabajar con el código fuente donde descomprimas linux-source.

cd /usr/src

El archivo fuente será algo como linux-source-5.15.0.tar.bz2

sudo tar -xf linux-source-5.15.0.tar.bz2

Crear un enlace simbólico a la carpeta descomprimida para facilitar el acceso

Puede que no haga falta

sudo ln -s linux-source-5.15.0 linux

Entrar al directorio de trabajo

cd linux

o

cd linux-source-5.15.0/

Al final, debes estar situado en el directorio donde está el archivo **Makefile**.

3. Carga la configuración del kernel actual (*make oldconfig)*

Copia la configuración del kernel que estás usando actualmente como punto de partida.

IMPORTANTE: ACUERDATE DE DESCOMPRIMIR EL ARCHIVO

sudo tar -xf linux-source-5.15.0.tar.bz2

Y luego posiciónate en el directorio que se crea (el que tiene el archivo Makefile).

Copia el archivo de configuración actual

uname -r

sudo cp /boot/config…. .config

o

sudo cp /boot/config-$(uname -r) .config

Comprueba: **ls -la | grep .config**

Carga la configuración, aplicando valores por defecto a nuevas opciones

Presiona Enter o ‘y’ para aceptar los valores por defecto si te pregunta.

sudo make oldconfig

4. Cuenta el número de componentes seleccionados para incluir directamente en el kernel (vmlinuz) o como módulos.

grep -E 'CONFIG_.*=(y|m)' .config | wc -l

o

Comando para conteo de módulos:

echo "Integrado (=y): $(grep -c '^[^#].*=y' .config)"

echo "Módulos (=m): $(grep -c '^[^#].*=m' .config)"

echo "Total: $(grep -c '^[^#].*=' .config)"

5. Ejecuta *make localmodconfig* esto ajustará la configuración solo a los módulos actualmente cargados en tu sistema (los que realmente usas).

Actualiza las opciones nuevas:

yes "" | make oldconfig

sudo make localmodconfig

La primera pregunta tiene "**no**" por defecto

La segunda pregunta usa "**N**".

6. Vuelve a contar el número de componentes configurados.

grep -E 'CONFIG_.*=(y|m)' .config | wc -l

o

Comando para conteo de módulos:

echo "Integrado (=y): $(grep -c '^[^#].*=y' .config)"

echo "Módulos (=m): $(grep -c '^[^#].*=m' .config)"

echo "Total: $(grep -c '^[^#].*=' .config)"

7. Realiza la primera compilación (*make -j <número de hilos> bindeb-pkg*) -- Sustituye <número de hilos> por el número de núcleos de tu CPU (consulta con nproc).

ATENCIÓN: antes de realizar la primera compilación:

https://gitlab.com/CalcProgrammer1/OpenRGB/-/issues/950

Usa nano en el archivo .config

Asegúrate que estos dos están así:

CONFIG_SYSTEM_TRUSTED_KEYS = ""

CONFIG_SYSTEM_REVOCATION_KEYS=""

Otros errores que pueden ocurrir es que no haya espacio suficiente de almacenamiento o poca RAM.

Usa "**nproc**" para saber cuántos núcleos tiene tu máquina.

Ahora usa: make -j <Número nucleos> bindeb-pkg

Ejemplo para **4** núcleos: **make -j 4 bindeb-pkg**

Si tienes algún error:

Limpiar la Compilación Anterior

Limpia todos los archivos de objetos, módulos y temporales

sudo make clean

Al final funciona. Recuerda, métele los comandos para que te detecte toda la capacidad del Disk (en plantillas de Antonio Carlos).

Y métele más RAM (Yo le he puesto 6GB).

sudo lvextend -l +100%FREE /dev/mapper/ubuntu--vg-ubuntu--lv

sudo resize2fs /dev/mapper/ubuntu--vg-ubuntu--lv

Cuando acabe el proceso de compulado, te debe salir algo así:

INSTALL debian/linux-libc-dev/usr/include

dpkg-deb: construyendo el paquete ` linux-libc-dev' en `../linux-libc-dev_5.15.189-2_amd64.deb'.

dpkg-deb: construyendo el paquete ` linux-image-5.15.189' en `../linux-image-5.15.189_5.15.189-2_amd64.deb'.

dpkg-deb: construyendo el paquete ` linux-image-5.15.189-dbg' en `../linux-image-5.15.189-dbg_5.15.189-2_amd64.deb'.

dpkg-genbuildinfo --build=binary -O../linux-upstream_5.15.189-2_amd64.buildinfo

dpkg-genchanges --build=binary -O../linux-upstream_5.15.189-2_amd64.changes

dpkg-genchanges: info: binary-only upload (no source code included)

dpkg-source --after-build .

dpkg-buildpackage: info: binary-only upload (no source included)

Los archivos se crean en el directorio superior al directorio que tiene el archivo Makefile.

8. Instala el núcleo resultando (paquete .deb) de la compilación, reinicia el equipo y comprueba que funciona adecuadamente.

Cuando termina el proceso anterior, se indica el nombre de los archivos ".deb" creados.

Ahora debemos instalarlo. En la guía que se puso en clase se indica que con el mismo comando se instale "linux-image" y "linux-header".

Ejemplo:

sudo dpkg -i linux-image-5.15.189_5.15.189-2_amd64.deb linux-headers-5.15.189_5.15.189-2_amd64.deb

Usa las fechas para saber que archivos usar.

Después:

sudo update-grub

sudo reboot

9. Si ha funcionado adecuadamente, utilizamos la configuración del paso anterior como punto de partida y vamos a reducir el tamaño del mismo, para ello vamos a seleccionar elemento a elemento.

- ***cp /boot/config-... .config #copia la configuración anterior***

Nos vamos al directorio:

cd /usr/src/linux-source-5.15.0/linux-source-5.15.0

cp /boot/config-5.15.189 .config

ls -la

- ***make clean*** *#limpia los archivos temporales*

- ***make xconfig*** *#interfaz grafica de configuracion*

Usa mejor la segunda opción:

Podemos instarlar:

a) **sudo apt install qtbase5-dev qtchooser qt5-qmake qtbase5-dev-tools**

   make xconfig #interfaz grafica de configuracion

o

b) **sudo apt install libncurses-dev**

   make menuconfig

10. Vuelve a contar los componentes

grep -E 'CONFIG_.*=(y|m)' .config | wc -l

o

Comando para conteo de módulos:

echo "Integrado (=y): $(grep -c '^[^#].*=y' .config)"

echo "Módulos (=m): $(grep -c '^[^#].*=m' .config)"

echo "Total: $(grep -c '^[^#].*=' .config)"

Ahora volvemos al punto 7:

i) make -j 4 bindeb-pkg

ii) E instala el paquete .deb generado.

sudo dpkg -i linux-image(el-nuevo).deb y lo que consideres oportuno.

Haz el proceso hasta que te quedes con **un kernel muy liviano** o se rompa la máquina.