Compilación del Kernel

Necesitamos el paquete build-essential.

 # apt-get install build-essential

Bajamos de la pagina www.kernel.org la fuente del kernel que queremos compilar :

 # cd /usr/src
 # axel -a -n 8 http://www.kernel.org/pub/linux/kernel/v3.0/linux-3.0.4.tar.bz2

o

 # wget http://www.kernel.org/pub/linux/kernel/v3.0/linux-3.0.4.tar.bz2

Descomprimimos el archivo :

 # tar xvf linux-3.0.4.tar.bz2
 # cd linux-3.0.4.tar.bz2

Hay veces que tenemos que insertar algún parche para nuestro kernel para hace tenemos que bajar el paquete patch

 # apt-get install patch

Luego para incluir el parche ejecutamos.

 # path -Np1 -i < ../Nombre-del-patch.patch

Veremos las opciones que le podemos pasar a ***make***

 # make help

```
...
i386_defconfig          - Build for i386
x86_64_defconfig        - Build for x86_64
…
```

Como vemos en el anterior ejemplo observamos podemos aplicar una configuración básica para ***i386*** (32 bits) o ***x86_64*** (64 bits).

Esta información la encontramos en ***arch/x86/configs***, como vemos en ***arch*** tendremos las distintas arquitecturas.

Si tenemos una compilación previa la limpiamos :

 # make mrproper

Si no tenemos ninguna configuración podemos ejecutar :

 # make i386_defconfig

o bien copiarnos una configuración de nuestro equipo que sabemos que realmente funciona :

 # cp /boot/config-2.6.32-5-amd64 .config

Si nuestro equipo es de 64bits y queremos compilar el núcleo para 32bits, exportamos la arquitectura a 32 bits.

 # export ARCH=i386

y controlamos si existen diferencias entre el kernel instalado y el nuevo haciendo :

 # make oldconfig

Luego si queremos seleccionar algún driver en particular tanto dentro del kernel o bien como modulo ejecutamos lo siguiente :

 # make menuconfig

Como veremos aquellos que están dentro del núcleo veremos que tiene un *******(asterisco) y aquellos que son modular es decir fuera del núcleo una letra ***M***(modular).

Una vez que terminamos salimos y grabamos y ejecutamos el make para que genere nuestra imagen.

 # make -j3

Compilamos los módulos

 # make modules

Esto lo que hace es instalar los módulos dentro de ***/lib/modules/3.0.4***

 # make modules_install

Y copiamos el archivo ***bzImage*** en ***/boot***.

 # cp arch/x86_64/boot/bzImage /boot/vmlinuz-3.0.4

Esto para 64 bits, si lo compilamos para 32 bits.

 # cp arch/x86/boot/bzImage /boot/vmlinuz-3.0.4
 # cp .config /boot/config-3.0.4
 # cp System.map /boot/System.map-3.0.4

Copiamos la documentación de esta versión del kernel.

 # install -d /usr/share/doc/linux-3.0.4
 # cp -r Documentation/* /usr/share/doc/linux-3.0.4

Podemos editar el archivo ***/boot/grub/grub.cfg*** y incorporamos el kernel compilado o bien ejecutamos el siguiente comando y automaticamente lo incorporara dentro de ***grub.cfg***.

 # update-grub2

o

 # update-grub

Esto automáticamente recorre el directorio ***/boot*** y agrega los kernel al ***grub.cfg***.

Para generar el ***initrd*** realizamos lo siguiente :

 # mkinitramfs -o initramfs-3.0.4 -k 3.0.4

-***o*** = Nombre del archivo de salida.
-***k*** = Esta opción especifica la versión del kernel.

Construir el paquete .deb kernel

Cuando empaquetamos el kernel como debian no tenemos que copiar nada, toda la información anterior estará dentro lo los paquetes que creamos, para crear el paquete del kernel como Debian debemos bajar el siguiente paquete :

 # apt-get install kernel-package

Vamos a usar el comando ***make-kpkg***. Este comando lo que hace básicamente es sustituir a las clásicas ***make dep***, ***make clean***, ***make bzImage*** y ***make modules***. Admite numerosas opciones y modificadores pero para lo que nos ocupa ahora sólo necesitaremos hacer lo siguiente:

 # make-kpkg clean
 # make-kpkg --initrd kernel_image kernel_headers

La opción --***initrd*** crea una imagen ***initrd*** en el paquete que se guardará en ***/boot*** cuando instalemos el kernel.

El comando anterior creara dos paquetes con extensión ***.deb*** en el directorio anterior. Un paquete será el ***kernel*** y el otro los ***kernel_headers***.

La opción de ***kernel_headers*** es opcional, se utiliza debido a que muchos programas y módulos necesitan tener los ***headers del kernel*** que se está usando para poder ser instalados, ejemplo: el driver de ***nvidia***, ***VMware***, ***ndiswrapper***, ***madwifi***, otros.

Si por alguna razón falla y dice que no se encontró el archivo *Documentation/lguest* realizamos lo siguiente para corregirlo :

 # mkdir Documentation/lguest
 # echo all: >> Documentation/lguest/Makefile

Para un QuadCore (*CONCURRENCY_LEVEL=5) = n^{a} procesadores + 1.*

 # INSTALL_MOD_STRIP=1 CONCURRENCY_LEVEL=5 fakeroot make-kpkg --initrd

--append-to-version=-edps --revision=+1.0 kernel_image kernel_headers

Instalación :

 # dpkg -i linux-headers-3.0.4_3.0.4-10.00.Custom_amd64.deb
 # dpkg -i linux-image-3.0.4_3.0.4-10.00.Custom_amd64.deb