**Recopilación de Kernel para Raspberry pi 2 y 3 de Elastix**

Hay dos métodos principales para la construcción del núcleo. Usted puede compilar en forma cruzada, que es mucho más rápido, pero requiere más configuración; o se puede construir de forma local en una raspberry pi que se llevará mucho tiempo.

**LA COMPILACIÓN CRUZADA**

En primer lugar se va a requerir un huésped Linux compilación cruzada adecuada.Tenemos la tendencia a usar Ubuntu; desde Raspbian es también una distribución Debian que significa el uso de líneas de comandos similares y así sucesivamente.

Usted puede hacer esto utilizando VirtualBox (o VMWare) en Windows, o instalarlo directamente en su ordenador. Como referencia se puede seguir las instrucciones en línea [en Wikihow](http://www.wikihow.com/Install-Ubuntu-on-VirtualBox) .

INSTALAR EL CONJUNTO DE HERRAMIENTAS

Utilice el siguiente comando:

* git clone https://github.com/RaspberryPi/herramientas

A continuación, puede copiar la cadena de herramientas a una ubicación común, como/ herramientas / brazo-bcm2708 / gcc-Linaro-brazo-linux-gnueabihf-raspbian, y añadir/ herramientas / brazo-bcm2708 / gcc-Linaro-brazo-linux-gnueabihf- raspbian / bina tu $ PATH en el .bashrc en su directorio personal. Para los sistemas host de 64 bits, utilice / herramientas / brazo-bcm2708 / gcc-Linaro-brazo-linux-gnueabihf-raspbian-x64 / bin. ¡Si bien este paso no es estrictamente necesario, que hace que sea más fácil para las líneas de comandos posteriores!

OBTENER FUENTES

Para obtener las fuentes, consulte el original [GitHub](https://github.com/raspberrypi/linux) repositorio para las diversas ramas.

* $ Git clone --depth = 1 https://github.com/raspberrypi/linux

CONSTRUIR FUENTES

Para construir las fuentes para la compilación cruzada que puede haber dependencias adicionales más allá de los que haya instalado por defecto con Ubuntu. Si usted encuentra que necesita otras cosas por favor enviar una solicitud de extracción para cambiar la documentación.

Introduzca los siguientes comandos para construir las fuentes y archivos de árbol de dispositivos.

Por Pi 1 o Compute Módulo:

* CD de Linux
* KERNEL = kernel
* hacer ARCH = brazo CROSS\_COMPILE = brazo - Linux - gnueabihf - bcmrpi\_defconfig

Por Pi 2/3:

* CD de Linux
* KERNEL = kernel7
* hacer ARCH = brazo CROSS\_COMPILE = brazo - Linux - gnueabihf - bcm2709\_defconfig

A continuación, por los dos:

* hacer ARCH = brazo CROSS\_COMPILE = brazo - Linux - gnueabihf - zImage módulos DTBS

Nota: Para acelerar la compilación en sistemas multiprocesador, y obtener una cierta mejora en los procesadores individuales, utilice -Jn donde n es el número de procesadores \* 1.5. ¡Alternativamente, siéntase libre de experimentar y ver lo que funciona!

INSTALAR DIRECTAMENTE EN LA TARJETA SD

Después de haber construido el núcleo que necesita para copiarlo en su RASPBERRY Pi e instalar los módulos; esto se hace mejor directamente a través de un lector de tarjetas SD.

En primer lugar el uso lsblk antes y después de conectar la tarjeta SD para identificar de cuál se trata; que debe terminar con algo como esto:

* sdb
  + sdb1
  + sdb2

Si se trata de una tarjeta de NOOBS debería ver algo como esto:

* sdb
  + sdb1
  + sdb2
  + sdb3
  + sdb5
  + sdb6

En el primer caso sdb1 / sdb5 es la partición FAT, y sdb2 / sdb6 es la imagen del sistema de ficheros ext4 (NOOBS).

Montarlos en primer lugar:

* mkdir mnt / FAT32
* mkdir mnt / ext4
* sudo mount / dev / sdb1 mnt / FAT32
* sudo mount / dev / sdb2 mnt / ext4

Ajustar los números de partición para las imágenes NOOBS.

A continuación, instalar los módulos:

* sudo make ARCH = brazo CROSS\_COMPILE = brazo - Linux - gnueabihf - INSTALL\_MOD\_PATH = mnt / ext4 modules\_install

Por último, copiar las gotas del núcleo y del árbol de dispositivos en la tarjeta SD, asegurándose de copia de seguridad de su antiguo núcleo:

* sudo cp mnt / FAT32 / $ NÚCLEO . img mnt / FAT32 / $ kernel-copia de seguridad . img
* secuencias de comandos sudo / mkknlimg arco / brazo / boot / zImage mnt / FAT32 / $ NÚCLEO . img
* sudo cp arch / brazo / boot / DTS / \* . DTB mnt / FAT32 /
* sudo cp arch / brazo / boot / DTS / superposiciones / \* . DTB \* mnt / FAT32 / superposiciones /
* sudo cp arch / brazo / boot / DTS / superposiciones / README mnt / FAT32 / superposiciones /
* sudo umount mnt / FAT32
* sudo umount mnt / ext4

Otra opción es copiar el kernel en el mismo lugar, pero con un nombre diferente - por ejemplo, kernel-myconfig.img - en lugar de sobrescribir el archivo kernel.img. A continuación, puede editar el archivo config.txt para seleccionar el kernel que el Pi se iniciará en:

* = kernel kernel-myconfig.img

Esto tiene la ventaja de mantener su núcleo se separan de la imagen del núcleo gestionado por el sistema y cualquier herramienta de actualización automática, y lo que le permite volver fácilmente a un núcleo de valores en el caso de que su núcleo no puede arrancar.

**CONSTRUCCIÓN LOCAL**

En una primera RASPBERRY Pi instalar la última versión de [Raspbian](https://www.raspberrypi.org/downloads) desde la página de descargas. Después arranca el pi, enchufe Ethernet para darle acceso a las fuentes, e ingrese.

En primer lugar, obtener las fuentes, que se llevará algún tiempo:

* git clone - profundidad = 1 https://github.com/RaspberryPi/Linux

Añadir dependencias que faltan:

* sudo apt - get install bc

Configurar el kernel -, así como la configuración por defecto es posible que desee[configurar el núcleo con más detalle](https://www.raspberrypi.org/documentation/linux/kernel/configuring.md) o [aplicar los parches de otra fuente](https://www.raspberrypi.org/documentation/linux/kernel/patching.md) para agregar o quitar funcionalidad requerida:

Ejecute los siguientes comandos dependiendo de la versión RASPBERRY Pi.

**RASPBERRY PI 1 (O COMPUTE MODULE) CONFIGURACIÓN POR DEFECTO DE CONSTRUCCIÓN**

* CD de Linux
* KERNEL = kernel
* hacer bcmrpi\_defconfig

RASPBERRY PI - COMPILACIÓN DE CONFIGURACIÓN POR DEFECTO 2/3

* CD de Linux
* KERNEL = kernel7
* hacer bcm2709\_defconfig

Construir e instalar el kernel, módulos y manchas árbol de dispositivos; este paso toma un largo tiempo ...

* hacer - J4 zImage módulos DTBS
* sudo make modules\_install
* sudo cp arch / brazo / boot / DTS / \* . DTB / boot /
* sudo cp arch / brazo / boot / DTS / superposiciones / \* . DTB \* / boot / superposiciones /
* sudo cp arch / brazo / boot / DTS / superposiciones / README / boot / superposiciones /
* secuencias de comandos sudo / mkknlimg arco / brazo / boot / zImage / boot / $ NÚCLEO . img

Nota: En un RASPBERRY Pi 2/3, el -j4 bandera se divide el trabajo entre los cuatro núcleos, acelerando la compilación de manera significativa.