EBook Gratis

APRENDIZAJE embedded-linux

Free unaffiliated eBook created from
Stack Overflow contributors.

#embedded -linux

Tabla de contenido

Acerca de 1

Capítulo 1: Empezando con incrustado de linux 2

Observaciones 2

Examples 2

Instalación o configuración 2

Creditos 6

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: embedded-linux

It is an unofficial and free embedded-linux ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official embedded-linux.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con incrustado de linux

Observaciones

Esta sección proporciona una descripción general de qué es el Linux incorporado y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande dentro de incrustado de Linux y vincular a los temas relacionados. Dado que la Documentación para incrustado en Linux es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

Examples

Instalación o configuración

Instrucciones detalladas sobre cómo configurar o instalar Linux incrustado.

ARM Versatile Express Emulation en Qemu

Introducción al medio ambiente:

Host Ubuntu: - 12.04

Versión del kernel de Linux: linux-4.4

versión busybox: 1.24.0

Cadena de herramientas de compilación cruzada: brazo-2014.05-29-arm-none-linux-gnueabi-i686-pc-linux gnu.tar.bz2

Versión qemu: qemu-2.5

Descarga e instalación de QEMU:

```
$ mkdir Source_Code
$ cd Source_Code
$ git clone git://git.qemu-project.org/qemu.git
$ cd qemu
$ git checkout remotes/origin/stable-2.5 -b stable-2.5
$ cd ../../
$ mkdir -p Binary_images/Qemu_Bin
$ cd Qemu_src/qemu
$ cd Source_Code/qemu
$ ./configure --target-list=arm-softmmu --prefix=/Path/to/your/Binary_images/Qemu_Bin
$ make
$ make install
```

Instalación de la cadena ARM Cross_Compiler Tool:

Descargar el código fuente: http://sourcery.mentor.com/public/gnu_toolchain/arm-none-linux-gnueabi/

Descargar -> arm-2014.05-29-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2

```
$ mkdir -p Binary_images/ARM_Cross_Tools
$ cd Binary_images/ARM_Cross_Tools
$ tar xvf arm-2014.05-29-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2
```

Descargue el código fuente del kernel de Linux:

```
$ cd Source_Code
$ git clone https://github.com/torvalds/linux
$ cd linux
# Switch to version v4.4
$ git checkout v4.4
```

Prepare la compilación: cargue la configuración predeterminada para la placa de destino, es decir, vexpress_defconfig.

```
$ make ARCH=arm CROSS_COMPILE=/path/to/your/Binary_images/ARM_Cross_Tools/arm-2014.05/bin/arm-none-linux-gnueabi- vexpress_defconfig
```

Ajustar o habilitar algunas configuraciones a partir de ahora no es necesario en el futuro, puede ser necesario

```
$ make ARCH=arm CROSS_COMPILE=path to your/Binary_images/ARM_Cross_Tools/arm-2014.05/bin/arm-none-linux-gnueabi- menuconfig
```

Compilar el kernel

```
$ make ARCH=arm CROSS_COMPILE=/path/to/your/Binary_images/ARM_Cross_Tools/arm-2014.05/bin/arm-none-linux-gnueabi- all
```

Verifique que qemu y el kernel puedan ejecutarse exitosamente:

```
~/Binary_images/Qemu_Bin/qemu-system-arm -M vexpress-a9 -m 512M -dtb
./arch/arm/boot/dts/vexpress-v2p-ca9.dtb -kernel ./arch/arm/boot/zImage -append
"console=ttyAMA0" -serial stdio
```

Compilando Busybox para ARM en QEMU:

Descargue Busybox desde https://busybox.net/downloads/

```
$ cd busybox
$ make ARCH=arm CROSS_COMPILE=/path/to/your/Binary_images/ARM_Cross_Tools/arm-2014.05/bin/arm-none-linux-gnueabi- defconfig
```

Habilitar o deshabilitar algunas configuraciones como se menciona a continuación

```
$make ARCH=arm CROSS_COMPILE=/path/to/your/Binary_images/ARM_Cross_Tools/arm-2014.05/bin/arm-none-linux-gnueabi- menuconfig
```

Configuración de Busybox -> Opciones de compilación ->

[*] Crea BusyBox como un binario estático (sin bibliotecas compartidas)

```
$ make ARCH=arm CROSS_COMPILE=/path/to/your/Binary_images/ARM_Cross_Tools/arm-2014.05/bin/arm-none-linux-gnueabi- install
```

El comando anterior construye Busybox y crea un directorio llamado _install que contiene el árbol del sistema de archivos raíz. A continuación, debe crear una carpeta para montar sistemas de archivos virtuales como proc, sys y scripts de inicio.

```
$ mkdir -p _install/proc/
$ mkdir -p _install/sys/
$ mkdir -p _install/tmp/
$ mkdir -p _install/root/
$ mkdir -p _install/var/
$ mkdir -p _install/mnt/
$ mkdir -p _install/etc/init.d/
```

Cree un nombre de archivo rcS dentro de la carpeta _install / etc / init.d / y edite el archivo rcS con el siguiente contenido

```
#!/bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin
runlevel=S
prevlevel=N
umask 022
export PATH runlevel prevlevel
mount -a
echo /sbin/mdev /proc/sys/kernel/hotplug
mdev -s
```

Cree un nombre de archivo inittab dentro de _install / etc / y edítelo con el contenido a continuación.

```
# /etc/inittab
::sysinit:/etc/init.d/rcS
console::askfirst:-/bin/sh
::ctrlaltdel:/sbin/reboot
::shutdown:/bin/umount -a -r
::restart:/sbin/init
```

Descargue y copie el archivo fstab a / etc /. carpeta en rootfs

```
$ wget clone
https://github.com/mahadevvinay/Embedded_Stuff/tree/master/Embedded_Linux_Virtual_Setup/fstab
```

Cree un archivo de imagen ext3 y copie todos los archivos en nuestra carpeta _install a la imagen:

```
$ dd if=/dev/zero of=RootFS.ext3 bs=1M count=$((32))
$ sudo mkfs.ext3 RootFS.ext3
$ mkdir tmpfs
$ sudo mount -t ext3 RootFS.ext3 tmpfs/ -o loop
$ sudo cp -r _install/* tmpfs/.
$ sudo umount tmpfs
```

El comando completo es emular:

```
~/Qemu_/Binary_images/Qemu_Bin/bin/qemu-system-arm -M vexpress-a9 -dtb path to your linux
folder/arch/arm/boot/dts/vexpress-v2p-ca9.dtb -kernel path to your linux
folder/arch/arm/boot/zImage -append root=/dev/mmcblk0 console=ttyAMA0 -sd path to your
busybox-1.24.0/RootFS.ext3 -serial stdio
```

La configuración anterior es para Qemu, se puede utilizar el mismo procedimiento para configurar cualquier objetivo incrustado.

Lea Empezando con incrustado de linux en línea: https://riptutorial.com/es/embedded-linux/topic/5479/empezando-con-incrustado-de-linux

Creditos

S. No	Capítulos	Contributors
1	Empezando con incrustado de linux	Community, Thiru Shetty, vinay hunachyal