EBook Gratuito

APPRENDIMENTO embedded-linux

Free unaffiliated eBook created from
Stack Overflow contributors.

#embedded -linux

Sommario

Di

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

Capitolo 1: Iniziare con embedded-linux

Osservazioni

Questa sezione fornisce una panoramica di cosa sia embedded-linux e perché uno sviluppatore potrebbe volerlo utilizzare.

Dovrebbe anche menzionare qualsiasi argomento di grandi dimensioni all'interno di embedded-linux e collegarsi agli argomenti correlati. Poiché la Documentazione per embedded-linux è nuova, potrebbe essere necessario creare versioni iniziali di tali argomenti correlati.

Examples

Installazione o configurazione

Istruzioni dettagliate su come installare o installare embedded-linux.

Emulazione Express versatile di ARM su Qemu

Introduzione dell'ambiente:

Host ubuntu: - 12.04

Versione del kernel Linux: linux-4.4

versione busybox: 1.24.0

Catena degli strumenti del compilatore incrociato: arm-2014.05-29-arm-none-linux-gnueabi-i686-pc-linux gnu.tar.bz2

versione qemu: qemu-2.5

Download e installazione di 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
```

ARM Cross_Compiler Tool chain installation:

Scarica il codice sorgente: http://sourcery.mentor.com/public/gnu_toolchain/arm-none-linux-gnueabi/

Scarica -> 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
```

Scarica il codice sorgente del kernel di Linux:

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

Preparare per la compilazione: carica config di default per la scheda di destinazione, ad esempio 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
```

Regola o abilita alcune impostazioni che in questo momento non sono necessarie in futuro potrebbero essere necessarie.

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

Compilare il kernel

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

Verifica che qemu e kernel possano essere eseguiti correttamente:

```
~/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
```

Compilazione di Busybox per ARM su QEMU:

Scarica Busybox da 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
```

Abilita o disabilita alcune impostazioni come indicato di seguito

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

Impostazioni Busybox -> Opzioni di creazione ->

[*] Costruisci BusyBox come file binario statico (nessuna librerie condivisa)

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

Il comando precedente crea Busybox e crea una directory chiamata _install contenente l'albero del filesystem di root. Successivamente, è necessario creare una cartella per il montaggio di file system virtuali come proc, sys e script di init.

```
$ 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/
```

Crea un nome di file rcS nella cartella _install / etc / init.d / e modifica il file rcS con contenuto sottostante

```
#!/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
```

Crea un nome di file inittab all'interno di _install / etc / e modificalo con il contenuto sottostante.

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

Scarica e copia il file fstab in / etc /. cartella in rootfs

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

Crea un file immagine ext3 e copia tutti i file nella nostra cartella _install per l'immagine:

```
$ 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
```

Il comando completo è quello di emulare:

```
~/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
```

L'impostazione sopra è per Qemu, la stessa procedura può essere utilizzata per configurare qualsiasi obiettivo incorporato.

Leggi Iniziare con embedded-linux online: https://riptutorial.com/it/embedded-linux/topic/5479/iniziare-con-embedded-linux

Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con embedded-linux	Community, Thiru Shetty, vinay hunachyal