

Introduzione eCos



Giovanni Perbellini



Agenda

- Introduzione eCos
- Toolchain
 - Download codice sorgente
 - Binutils
 - GCC/G++
 - Newlib
 - GDB/Insight
 - Compilazione toolchain
 - Selezione Target
- Compilazione eCos
 - Configtool (modalità grafica)
 - ecosconfig (modalità testuale)
- Compilazione ed esecuzione programmi
- Esercizi: Helloworld, Thread

Introduzione eCos (I)

- Sistema operativo embedded Real-Time (open source - GPL license)
 - <http://ecos.sourceware.org>
- Progettato per dispositivi con dimensione della memoria di qualche centinaia di kilobyte
- Carica su una larga varietà di piattaforme HW: ARM, MIPS, PowerPC, etc.
- Il kernel fornisce thread scheduling, synchronization, timer e primitive di comunicazione

Introduzione eCos (II)

- Gestisce risorse HW
 - interrupts, exceptions, memory, caches
- Supporto una larga varietà di dispositivi
 - serial devices, ethernet controllers e memorie FLASH
- Contiene uno stack TCP/IP
- Hardware Abstraction Layer (HAL)
 - Nasconde le caratteristiche specifiche delle CPU e piattaforme (ARM, MIPS, Intel x86, PowerPc, etc.)

Download codice sorgente toolchain

- GNU binary utilities
 - Include il GNU assembler (as) e il GNU linker (ld)
- GNU compiler collection core
 - Il compilatore C (gcc)
- GNU compiler collection g++
 - Il compilatore C++ (g++)
- Newlib
 - La libreria C per sistemi embedded
- Insight or GDB
 - GNU debugger

Target

- Va specificata l'architettura target per cui si vuole compilare

ARCHITECTURE	TARGET
ARM (including StrongARM, XScale)	arm-elf
Intel x86	i386-elf
Matsushita AM3x	mn10300-elf
Motorola 68K/ColdFire	m68k-elf
MIPS32	mipsisa32-elf
NEC V850	v850-elf
NEC VR4300	mips64vr4300-elf
PowerPC	powerpc-eabi
Renesas H8/300H	h8300-elf
Renesas SuperH	sh-elf
Toshiba TX39	mips-tx39-elf
Toshiba TX49	mips-tx49-elf

Compilazione/Installazione binutils

- `mkdir -p /tmp/user/work/binutils`
- `cd /tmp/user/work/binutils`

- `/tmp/user/src/binutils-2.13.1/configure`
 `--target=powerpc-eabi`
 `--prefix=/tmp/user/crosscomp`

- `make all`
- `make install`

NewLib

- Insieme di librerie di supporto per dispositivi embedded
 - Funzioni matematiche
 - Funzioni per input/output
 - Funzioni per gestire le stringhe
 - ...
- `mv newlib-1.11.0/newlib gcc-3.2.1`
- `mv newlib-1.11.0/libgloss gcc-3.2.1`

Compilazione/Installazione GCC/G++

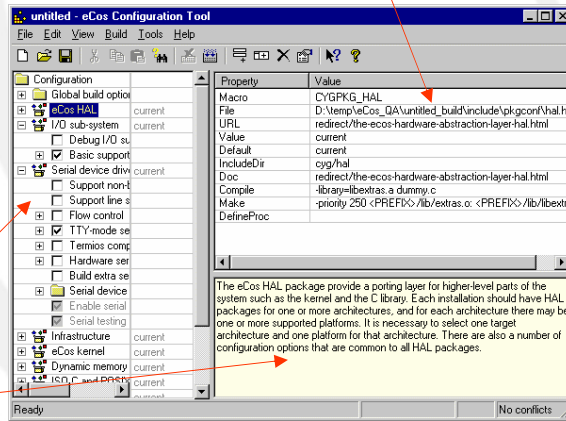
- `PATH=/tmp/user/crosscomp/bin:$PATH`
`export PATH`
`mkdir -p /tmp/user/work/gcc`
`cd /tmp/user/work/gcc`
- `/tmp/user/src/gcc-3.2.1/configure`
`--target=powerpc-eabi`
`--prefix=/tmp/user/crosscomp`
`--enable-languages=c,c++`
`--with-gnu-as --with-gnu-ld --with-newlib`
`--with-gxx-include-dir=/tmp/user/crosscomp/powerpc-eabi/include`
- `make all`
- `make install`

Compilazione/Installazione GDB

- `mkdir -p /tmp/user/work/gdb`
`cd /tmp/user/work/gdb`
- `/tmp/user/src/gdb-6.6/configure`
`--target=powerpc-eabi`
`--prefix=/tmp/user/crosscomp`
`--enable-sim-powerpc`
`--enable-sim-stdio`
- `make all`
- `make install`

eCos Configuration tool (I)

- Composto da 5 finestre
 - Selezione del TARGET
 - Compilazione sorgenti
 - Linux/Windows version



eCos Configuration tool (II)

- Configuration Window
 - Selezione dei moduli del sistema operativo
- Conflicts Window
 - Visualizza i conflitti della configurazione selezionata

Item	Conflict	Property
CYGSEM_HAL_STATIC_MMU_TABLES	Unsatisfied	Requires CYGSEM_HAL_INSTALL_M...

Item	Conflict	Property
CYGSEM_HAL_STATIC_MMU_TABLES	Unsatisfied	Requires CYGSEM_HAL_INSTALL_M...

eCos Configuration tool (II)

- Output Window
 - Visualizza tutti i risultati della compilazione
- Properties Window
 - Visualizza le proprietà di un *item* selezionato nella *Configuration Window*
- Short description Window
 - Visualizza una breve descrizione di un *item* selezionato nella *Configuration Window*

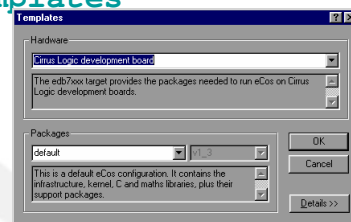
Property	Value
URL	ref/ecos-ref/the-ecos-hardware-abstraction-layer-hal.html
File	include/glibconfig/hal.h
Macro	C:\GPKG_HAL
Value	v1_3
Default	v1_3
Type	String
IncludeDir	osp/hal
Doc	ref/ecos-ref/the-ecos-hardware-abstraction-layer-hal.html
Compile	generic_stub.c thread-packets.c hal_stub.c div_api.c
Compile	library-libewas.a dlistmp.c

Compilazione eCos – configtool (I)

- Repository
 - La gerarchia di directory che contiene i sorgenti e altri file di supporto usati per costruire una configurazione eCos
 - Selezionare: **Build** → **Repository**
- Toolchain
 - Build tool (compiler, linker, etc.)
 - Selezionare: **Tool** → **Paths** → **Build Tools**
 - User tool (cat, ls, etc.)
 - Selezionare: **Build** → **Paths** → **User Tools**

Compilazione eCos – configtool (II)

- Templates
 - Permettere di scegliere per quale Hardware (Powepc, ARM9, MIPS, etc.) generare una configurazione eCos
 - Selezionare: **Build** → **Templates**
 - Con questa operazione si possono generare alcuni conflitti
 - Risoluzione automatica



Compilazione eCos – configtool (III)

- Generazione di una configurazione eCos
 - Compilazione
 - Selezionare: **Build** → **Library**
 - I risultati della compilazione sono visualizzabili nella *Output Window*
 - Richiesta di salvataggio file di configurazione (e.g. *powerpc-default.ecc*)
 - In alternativa selezionare: **File** → **Save**

Compilazione eCos – configtool (IV)

- Generazione della configurazione eCos
 - Generazione della directory “build”
 - Contiene tutti i file generati: makefile, object files, file di dipendenze, etc.
 - e.g. `powerpc-default_build`
 - Generazione della directory “install”
 - Contiene tutti i file necessari per lo sviluppo di applicazioni (libreria libtarget.a e header files)
 - e.g. `powerpc-default_install`

Compilazione eCos – ecosconfig (I)

- Impostare le variabili di ambiente:
 - PATH

```
$ PATH=/usr/local/ecos/ecos-2.0/tools/bin:$PATH
$ export PATH
```
 - ECOS_REPOSITORY

```
$ ECOS_REPOSITORY=/usr/local/ecos/ecos-2.0/packages
$ export ECOS_REPOSITORY
```
- Creare la directory di lavoro dove installare eCos Synthetic Target

```
$ mkdir /tmp/user/ecos/work-powerpc
$ cd /tmp/user/ecos/work-powerpc
```

Compilazione eCos – ecosconfig (II)

- ecosconfig Help
 - `$ ecosconfig --help`
- Creazione di un nuovo target (e.g.: Power-pc)
 - `$ ecosconfig new <target>`
 - Per Power-pc Target `<target> = psim`
- Costruzione albero delle directory con sorgenti di eCos
 - `$ ecosconfig tree`
- Compilazione eCos
 - `$ make`

Compilazione applicazioni

- Compilazione (regola generale)
 - `$ TARGET-gcc -c -IINSTALL_DIR/include file.c`
 - `$ TARGET-gcc -o program file.o`
 - LINSTALL_DIR/lib
 - Ttarget.ld -nostdlib
- Esempio
 - `$ powerpc-eabi-gcc -c -IINSTALL_DIR/include -I.`
 - ffunction-sections -fdata-sections
 - g -O2 file.c
 - `$ powerpc-eabi-gcc -o program file.o -ffunction-`
 - sections -fdata-sections -Wl,--gc-sections -g
 - O2 -LINSTALL_DIR/lib -Ttarget.ld -nostdlib

Esecuzione applicazioni

- Esecuzione
 - Run (TARGET = **powerpc-eabi**)
 - TARGET-run program
 - Debug (TARGET = **powerpc-eabi**)
 - TARGET-gdb program
 - (gdb) target sim
 - (gdb) load
 - (gdb) break main
 - (gdb) run
- Esecuzione con *console* simulata (utilizzo di PSIM)
 - `powerpc-eabi-run -o '/iobus/pal@0xf0001000/reg 0xf0001000 32` program`
 - Opzione per definire una porta di I/O nello spazio di memoria del simulatore usato da eCos per la console di output

Esercizi

- Helloworld
- Thread