

# uARM: a simple ARM Virtual Machine

Marco Melletti

17/12/2013

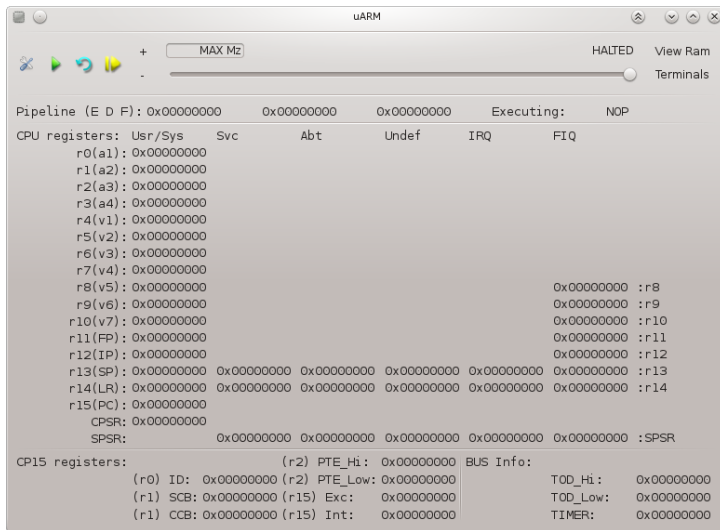
- Chip : PDP-11, device ancora utilizzati (1983)
- MPS : MIPS, memoria virtuale sempre attiva (2004)
- uMPS : MIPS, memoria virtuale opzionale (2007)
- uMPS 2 : MIPS, supporto multicore, interfaccia grafica ristrutturata (2011)
- uARM

- Architettura RISC: Reduced Instruction Set Computer
- Attuale e largamente utilizzata:
  - ▶ Embedded Systems
  - ▶ Smartphones
  - ▶ Nintendo DS
  - ▶ Raspberry Pi
  - ▶ Game Boy Advance
  - ▶ iPod
  - ▶ ...

- Processore ARM7TDMI
- Memoria Little-Endian a dimensione variabile
- 8 Devices per tipo:
  - ▶ terminali
  - ▶ stampanti
  - ▶ schede di rete (VDE)
  - ▶ dischi fissi
  - ▶ nastri (dischi ottici)

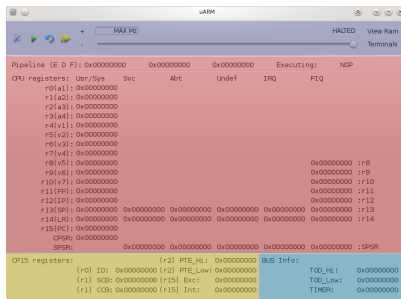
# uARM: GUI

Si presenta così:



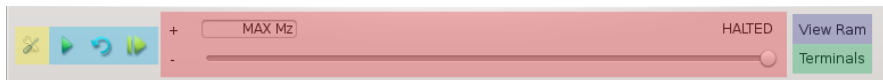
# uARM: GUI

Quattro aree principali:



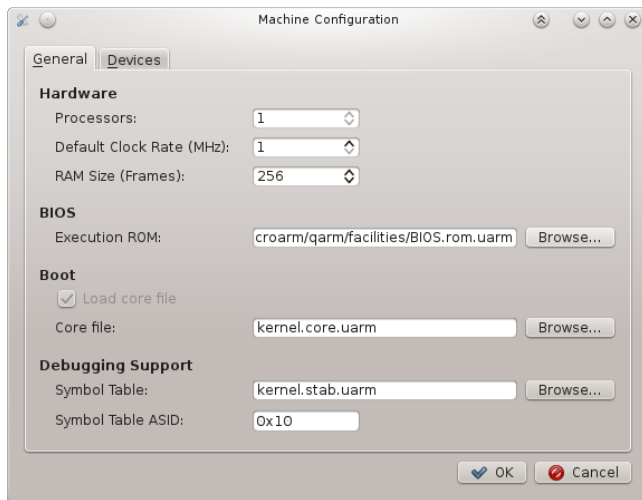
- Barra di controllo
- Vista Processore
- Vista Coprocessore
- Vista Informazioni di Sistema

# uARM: Barra di controllo



- Configurazioni
- Controllo esecuzione
- Velocità emulazione
- Ram Inspector
- Terminali

# uARM: Finestra Configurazioni



The image shows a 'Machine Configuration' dialog box with two tabs: 'General' and 'Devices'. The 'General' tab is active. It contains sections for 'Hardware', 'BIOS', 'Boot', and 'Debugging Support'. The 'Hardware' section has three spinners: 'Processors' (1), 'Default Clock Rate (MHz)' (1), and 'RAM Size (Frames)' (256). The 'BIOS' section has a text field for 'Execution ROM' containing 'croarm/qarm/facilities/BIOS.rom.uarm' and a 'Browse...' button. The 'Boot' section has a checked 'Load core file' checkbox, a text field for 'Core file' containing 'kernel.core.uarm', and a 'Browse...' button. The 'Debugging Support' section has a text field for 'Symbol Table' containing 'kernel.stab.uarm' and a 'Browse...' button, and a text field for 'Symbol Table ASID' containing '0x10'. At the bottom are 'OK' and 'Cancel' buttons.

Machine Configuration

General Devices

**Hardware**

Processors: 1

Default Clock Rate (MHz): 1

RAM Size (Frames): 256

**BIOS**

Execution ROM: croarm/qarm/facilities/BIOS.rom.uarm Browse...

**Boot**

☒ Load core file

Core file: kernel.core.uarm Browse...

**Debugging Support**

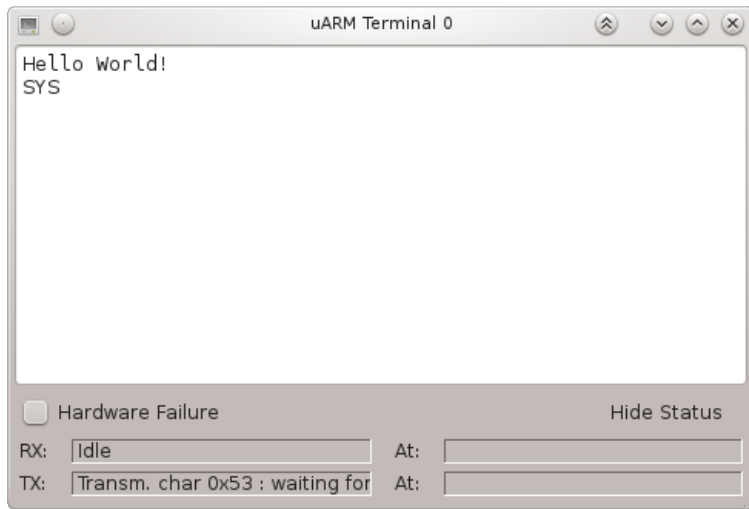
Symbol Table: kernel.stab.uarm Browse...

Symbol Table ASID: 0x10

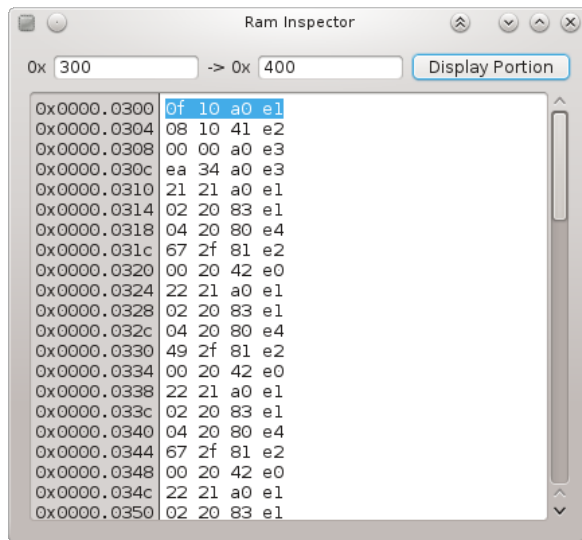
OK Cancel



# uARM: Finestra Terminale



# uARM: Finestra Ram Inspector



# uARM: Un esempio

/usr/include/uarm/test/helloWorld.c:

```
#include "../facilities/libuarm.h"
```

```
char *saluto = "Hello World!\n\0";
```

```
char *strano = "If this got printed something fishy has happened..\n\0";
```

```
int main(){
```

```
    tprint(saluto);
```

```
    HALT();
```

```
    tprint(strano);
```

```
    return 0;
```

```
}
```

# uARM: Un esempio

Compiliamo il file:

```
$> arm-none-eabi-gcc -mcpu=arm7tdmi -c -o helloWorld.o \  
    /usr/include/uarm/test/helloWorld.c
```

Linkiamolo con le librerie di uARM:

```
$> arm-none-eabi-ld -T \  
    /usr/include/uarm/ldscripts/elf32ltsarm.h.uarmcore.x \  
    -o helloWorld /usr/include/uarm/crtso.o \  
    /usr/include/uarm/libuarm.o helloWorld.o
```

E per concludere convertiamolo nel formato di uARM:

```
$> elf2uarm -k test/helloWorld.elf
```

E adesso vediamo in azione...

- Home page: <http://mellotanica.com/uarm>
- Non ha ancora il set completo di funzionalità...
- ... ma per febbraio le avrà!
- Contattatemi per domande/problemi/richieste
- [marco.melletti@studio.unibo.it](mailto:marco.melletti@studio.unibo.it)

Grazie dell'attenzione

Ci sono domande?