

Sistemas Embebidos I

Ano Letivo 2014/2015 - Semestre de Inverno

Lab1 - Ambiente de trabalho

1 Introdução

No desenvolvimento de aplicações para sistemas dedicados, tipicamente, existem duas máquinas envolvidas: o *host* e o *target*. O *host* é normalmente uma estação de trabalho genérica, com SO Windows, Linux ou outro, em que estão instaladas as ferramentas para o desenvolvimento das aplicações a executar no *target*. Quanto ao *target*, consiste do dispositivo alvo para as aplicações a desenvolver e que tem uma arquitetura diferente da do *host*, por exemplo, um microcontrolador com arquitetura ARM.

Face a esta diversidade, o ambiente de trabalho depende obviamente do tipo de *host* e do tipo de *target* considerados. Na unidade curricular Sistemas Embebidos I (SE1) será utilizado como *target* o microcontrolador LPC2106 da NXP, que tem arquitetura ARM v7, e como *host* um PC com SO Linux sob a forma de uma máquina virtual.

O ambiente de trabalho será baseado nas ferramentas da GNU para gerar o código executável para o *target*, e no *debugger* Insight para a depuração desses programas. Este *debugger* é baseado no GDB e tem uma interface gráfica que facilita a sua utilização.

2 Preparação do ambiente de trabalho no *host*

2.1 Instalação do *host*

O *host* consiste de uma máquina virtual VMWare, cujo SO é uma distribuição ubuntu com uma interface gráfica simples (XFCE). Para a sua execução deve ser utilizado o VMWare Player. A imagem da máquina virtual e o VMWare Player estão disponíveis na secção “Ferramentas” da página da unidade curricular SE1 na plataforma Thoth.

2.2 Mudança do *layout* do teclado para Português

No VMWare Player, deverá seleccionar as seguintes opções para configurar a máquina virtual do *host*:

1. Seleccionar **Settings Manager -> Keyboard -> Layout** e remover a selecção: **Use system defaults**.
2. Seleccionar **Keyboard Layout -> Add -> Portuguese -> Portuguese (eliminate dead keys)**
3. Remover a entrada **English (US)**.

2.3 Definição de uma diretoria partilhada

No VMware Player, deverá adicionar uma diretoria partilhada para que seja possível ao *host* aceder ao sistema de ficheiros do computador onde a máquina virtual é executada. Para tal, selecione **Virtual Machine -> Virtual Machine Settings -> Options -> Shared Folders -> Add** e seguidamente indique o nome da diretoria partilhada e a sua localização na máquina *host*

Name: host-sel
Host Path: <nome da diretoria na máquina host>

2.4 Atalho para a diretoria partilhada

Para criar um atalho no ambiente de trabalho do *host* para a diretoria partilhada, abra um terminal

(janela de comando) e execute o seguinte comando:

```
$ ln -s /mnt/hgfs/host-se1 ~/Desktop
```

2.5 Instalação de utilitários

Para instalar o *geany*, o editor de texto que servirá de base ao desenvolvimento de aplicações para a arquitetura ARM, abra um terminal (janela de comando) e execute o seguinte comando:

```
$ sudo apt-get install geany
```

2.6 Instalação das ferramentas ARM

Para instalar no *host* as ferramentas GNU para desenvolvimento de aplicações para a arquitetura ARM, comece por criar uma diretoria temporária (por exemplo **tmp_tools**) dentro da diretoria **user** da máquina virtual. Copie para esta diretoria os seguintes ficheiros da secção “Ferramentas” na página da unidade curricular SE1 na plataforma Thoth :

```
arm-2009q1-161-arm-none-eabi-i686-pc-linux-gnu.tar.bz2
insight-6.8-1a.tar.bz2
insight-texi.patch
```

De seguida, abra um terminal (janela de comando) nesta diretoria e execute os seguintes comandos para instalar no *host* as ferramentas da GNU para a arquitetura ARM:

```
$ sudo mkdir /usr/local/arm-tools
$ sudo tar xvf arm-2009q1-161-arm-none-eabi-i686-pc-linux-gnu.tar.bz2
-C/usr/local/arm-tools
$ geany ~/.bashrc
```

Acrescente as seguintes alterações no final do ficheiro:

```
#PATH arm-none-eabi-...
PATH=$PATH:/usr/local/arm-tools/arm-2009q1/bin
```

Terminar gravando as alterações.

Execute agora os seguintes comandos para instalar no *host* uma versão do *debugger* Insight orientada ao desenvolvimento de aplicações para a arquitetura ARM:

```
$ sudo apt-get install texinfo libncurses5-dev libx11-dev libexpat1-dev
libusb-dev libftdi-dev
$ tar xvf insight-6.8-1a.tar.bz2
$ cd insight-6.8-1
$ patch -p1 < ../insight-texi.patch
$ mkdir build
$ cd build
$ ../configure --target=arm-none-eabi \
               --prefix=/usr/local/arm-tools/arm-2009q1 \
               --enable-languages=c,c++ \
               --enable-thumb \
               --enable-interwork \
               --enable-multilib \
               --enable-tui \
               --with-newlib \
               --disable-werror \
               --disable-libada \
               --disable-libssp \
               --with-expat
```

```
$ make
$ sudo make install
```

No final deste processo pode remover a diretoria temporária que criou para a instalação do ambiente de trabalho.

3 Diretorias de trabalho

Para a organização do ambiente de trabalho, propõe-se que utilize a seguinte estrutura de diretorias:

```
sel
|
+-- bsp
+-- docs
+-- includes
+-- libs
+-- sources
|   +-- exercicio_1
|       +-- debug
|       +-- release
|   +-- exercicio_2
|       +-- debug
|       +-- release
|   +-- ...
```

A diretoria **bsp** (*board supported packages*) serve para alojar os ficheiros de configuração das ferramentas (por exemplo, os scripts de configuração do openocd e do linker).

A diretoria **docs** serve para alojar os relatórios dos exercícios realizados em laboratório, bem como outros documentos relevantes para a realização dos trabalhos de laboratório e do projeto (por exemplo, os enunciados do trabalhos, os datasheets dos periféricos, etc.).

A diretoria **includes** serve para alojar os ficheiros que contêm os protótipos dos diferentes módulos que serão desenvolvidos ao longo do semestres nos diferentes exercícios de laboratório.

A diretoria **libs** guarda as bibliotecas que podem ser utilizadas pelas aplicações a ser desenvolvidas ao longo do semestre nos trabalhos de laboratório e no projeto.

Sob a diretoria **sources** deverão ser criadas várias diretorias, uma para cada exercício de laboratório (**exercicio_1**, **exercicio_2**, ...) e outra para o projeto (**proj**), onde serão guardados os ficheiros fonte das aplicações e dispositivos a desenvolver. Cada diretoria deverá ainda conter duas subdiretorias (**debug** e **release**), em que deverão ser alojados os ficheiros executáveis e os ficheiros intermédios das aplicações desenvolvidas. A diretoria **debug** respeita à versão para execução na RAM do *target*, enquanto que a diretoria **release** respeita à versão final da aplicação, que será executada a partir da memória FLASH do *target*.

4 Utilização das ferramentas GNU

4.1 Compilação

Para transformar as instruções de mnemónicas legíveis para o código máquina respetivo da arquitetura ARM deve usar o *assembler* **as** da GNU. A sintaxe do comando para compilar um ficheiro fonte em linguagem *assembly* é a seguinte:

```
$ arm-none-eabi-as <nome do ficheiro fonte>.S -o <nome do ficheiro objeto>.o
```

4.2 Localização em memória

Este processo permite localizar o programa em endereços de memória válidos para o *target*. A definição destes endereços é explicitada num ficheiro de definições para a ferramenta **ld** da GNU.

Atendendo a que nesta fase inicial dos trabalhos apenas se vai utilizar o emulador da arquitetura ARM para executar os programas desenvolvidos, a ferramenta **ld** pode funcionar com as suas definições por omissão. Nesta situação, para se gerar um ficheiro executável é apenas necessário definir o símbolo de entrada do programa, que deve ter o nome `_start`.

A sintaxe do comando para obter o ficheiro final localizado em memória é a seguinte:

```
$ arm-none-eabi-ld <nome do ficheiro objeto>.o -o <nome do ficheiro loc>.axf
```

Para verificar o resultado dos dois comandos anteriores, pode ser utilizado o utilitário **objdump**. A sintaxe deste comando é a seguinte:

```
$ arm-none-eabi-objdump -S <nome do ficheiro objeto>.o
```

```
$ arm-none-eabi-objdump -S <nome do ficheiro loc>.axf
```

4.3 Teste / debug

Para testar o programa é utilizado o *debugger* Insight, que é lançado no terminal com o comando

```
$ arm-none-eabi-insight <nome do ficheiro loc>.axf
```

Para utilizar o simulador/emulador disponibilizado pelo GDB deve configurar o Insight desta forma:

1. Selecionar **File -> Target Settings** e de seguida a opção **Target: Simulator**
2. Selecionar a opção **Set breakpoint at _start**.