Using the C Standard Library with ARMSim#

Important Note

Some components of the Mentor Graphics Sourcery CodeBench are needed if access to the standard C library is required or if C source code is to be compiled. Unfortunately Mentor Graphics has stopped distributing the free Lite Edition of the CodeBench for the ARM EABI (the platform we need). Only a professional version is currently available. The professional edition may be obtained from this URL:

`http://www.mentor.com/embedded-software/sourcery-tools/sourcery-code-bench/editions/lite-edition/`

Look on the webpage for the download named *Sourcery CodeBench for ARM Embedded*.

The particular library archive files we need for running C programs are, however, included with this ARMSim# distribution. (Redistributing the files does not infringe any laws because these files were extracted from a distribution to which the GNU General Public License applies.)

The Sourcery CodeBench version of the gcc compiler is not included with the ARMSim# distribution due to its size.

Instructions for Windows Systems

1. Compile any C source files to generate ARM binary files.

   A gcc compiler which cross-compiles to the ARM architecture must be used.

   For example, if there is one file named `factorial.c` then the command which uses the Sourcery CodeBench cross-compiler is

   ```
   arm-none-eabi-gcc -c factorial.c
   ```

   [Alternatively, you can write an ARM assembly language program which calls functions in the C library. However, you need to follow the linkage conventions when calling these functions.]

2. Start ARMSim#

   Check File/Preferences to make sure that the Angel SWI instruction set is enabled as a plugin. (It should be enabled by default.)

   Open File/Load Multiple and select the files and libraries listed below.

   Begin with any `.o` files produced in step 1 above.

   Add any ARM assembler files to be included in the program.

   Then add these four:

   ```
   start.o
   libc.a
   libcs3hosted.a
   libgcc.a
   ```

   Note: if you installed the Sourcery CodeBench, then the last three of these files may be found in these locations:

```
PATH_TO_CODEBENCH\arm-none-eabi\lib\libc.a
PATH_TO_CODEBENCH\arm-none-eabi\lib\libcs3hosted.a
PATH_TO_CODEBENCH\lib\gcc\arm-none-eabi\4.8.3\libgcc.a
```

where *PATH_TO_CODEBENCH* is the location of the folder where the Mentor Graphics Sourcery CodeBench was installed on your computer. The full path would likely be similar to this if the package was installed in a user's Documents folder:

```
C:\Users\USERNAME\Documents\MentorGraphics\
Sourcery_CodeBench_Lite_for_ARM_EABI
```

The `start.o` file and its assembler source file `start.s` are provided with the distribution.

Either a C file or an ARM assembly file must define a function named `main` which accepts two arguments: an integer and an array of strings. This emulates the main entry point of a C program on Unix/Linux. (However the provided version of `start.o` always passes 1 as the integer and the array contains the single string `"a.out"`.)

If there are no errors reported in the Console window, then the ARM program is ready to run. Execution begins in the `start.o` file, but the code in that file does little more than invoke the program's `main` function.

Running ARM programs with Code Sourcery's ARM simulator

The Mentor Graphics Sourcery CodeBench once included a separate simulator that executed an ARM program (`arm-none-eabi-run`).

1. An example compile ccommand to create an executable program is

   ```
   arm-none-eabi-gcc -c factorial.c -o factorial -T generic-hosted.ld
   ```

   Executable programs can be created for the ARM from other sources of course. The important detail to note is that the final link step needs the `-T generic-hosted.ld` options, where `generic-hosted.ld` is a file provided with the Sourcery CodeBench.

2. To run the program created as the ELF format executable file named `factorial` in the example, use the command

   ```
   arm-none-eabi-run factorial
   ```

Later distributions do not provide this simulator. Instead the version of `gdb` provided with the Sourcery CodeBench can be used. Usage is as follows.

1. Create the executable program in the same way as step #1 above.
2. Start the CodeSourcery version of `gdb` on the program

   ```
   arm-none-eabi-gdb factorial
   ```

3. Enter these three `gdb` subcommands

   ```
   target sim
   load
   run
   ```