IMPLEMENTING SPSCL PROGRAMS

Using Codeblocks

Dr. José M. Garrido
Department of Computer Science

Jan, 2022

College of Computing and Software Engineering
Kennesaw State University

1 Introduction

There are several Integrated Development Environments (IDEs) that help implementation of programs written in C, C++, Fortran, and other programming languages. Eclipse is one of the most complete and powerful tools, it is mainly useful for Java programs and in general, it can be very slow. For C++, C, and Fortran programs, Codeblocks and CodeLite are faster, lighter, and more convenient to use.

This document briefly explains implementing SPSCL programs using CodeBlocks on Linux. For a more detailed introduction to CodeBlocks, refer to various websites with CodeBlocks tutorial and documentation.

2 CodeBlocks Settings for SPSCL

The following sequence of steps involve the basic procedure for setting the appropriate options on Codeblocks for using the SPSCL translator then the remaining steps for editing C programs, compiling, and linking.

2.1 Download the SPSCL Translator

The preliminary step required is download and the installation of the the SPSCL translator. The 64-bit version of SPSCL translator is available on the following web page:

`ksuweb.kennesaw.edu/~jgarrido/scl`

1. Download the archive file `scl_linux.tar.gz` from the web page and extract all files from archive.
2. Select the executable file of the SPSCL translator (`scl.out`). This executable file, the `scl.out` and the `scl.h` file must be stored on a folder such as `~/SPSCL`.

2.2 Setting the Codeblocks Options

1. Start CodeBlocks, the screen that appears is shown in Figure 1. Click on 'Create a new project'.

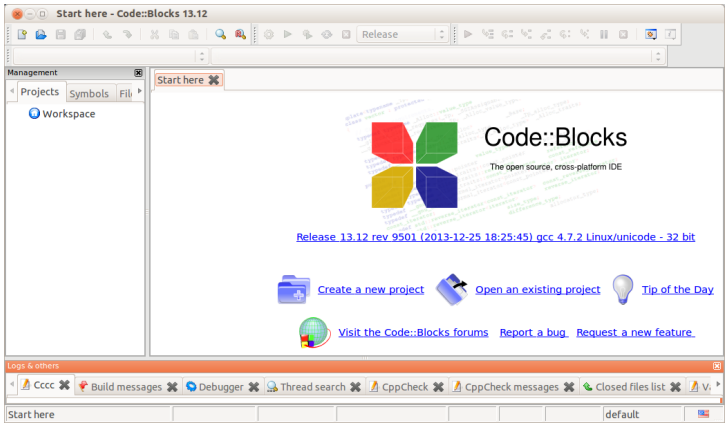

Figure 1: Starting Codeblocks.

2. On the top bar click on the Settings menu, then select Compiler. A dialog box appears. On the tab group, activate the Other Options tab, and click the button Advanced options located on the lower right of the dialog box. See Figure 2.

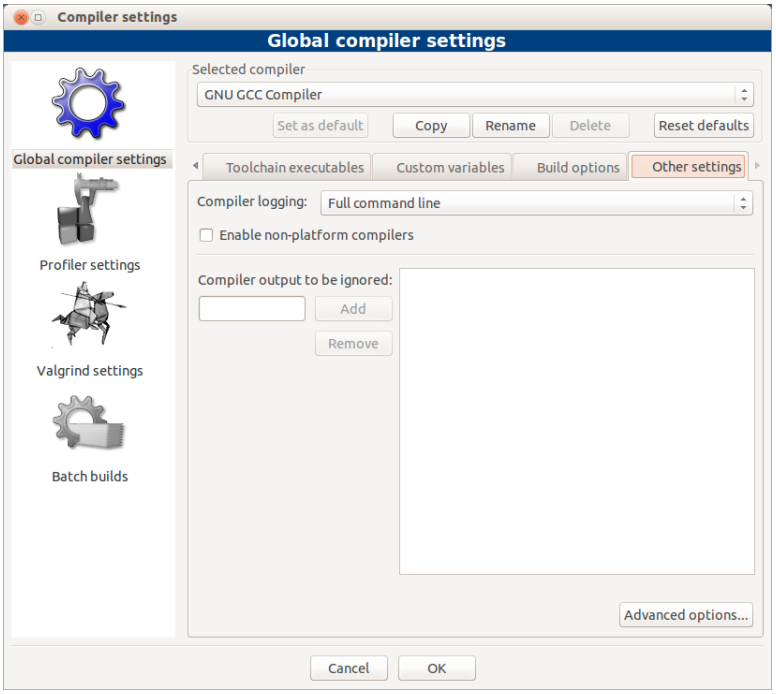

Figure 2: Selecting Advanced options.

3. Click the Yes button when the box *Edit advanced compiler setting?* appears.

4. On the new window that appears, click the '+' button and type `scl` in the dialog box.

5. Type the Command line macro and the Generated files. The window will now appear as shown in Figure 3. This assumes that the SPSCL translator (executable file `scl`) is locate in folder: `~/SPSCL`

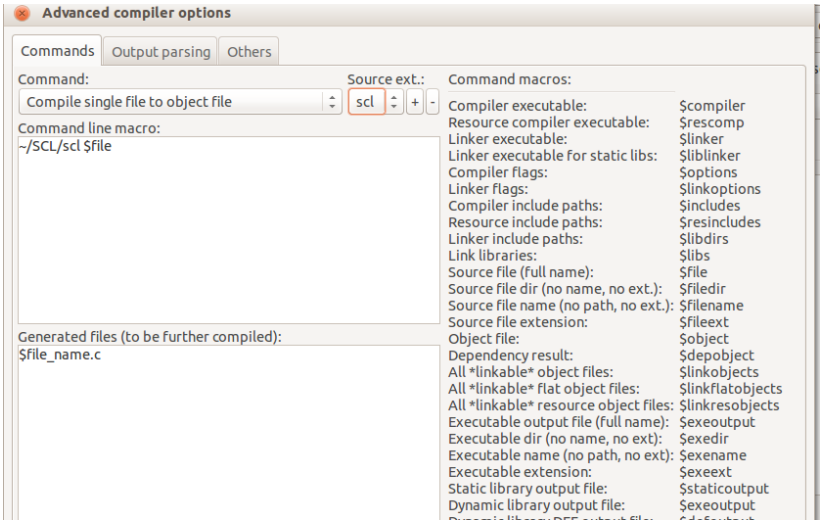

Figure 3: Adding the SPSCL extension and command for translating SPSCL files.

6. Click the Ok button located on the bottom of the window.

Codeblocks is now setup to recognize source files with an `scl` extension for editing and invoking the SPSCL translator.

2.3 Using Codeblocks for Implementing SPSCL Programs

1. Select Console Application and click the Go button, which is located on the upper right corner of the window.

2. On the Console Application window, click the Next button. Select C language and click Next.

3. Type the project title (name), e.g. *Welcome*. In this example, the project will be created in folder: `/home/jgarrido/comp_progs/scl_progs`, as shown in Figure 4. Click the Next button.

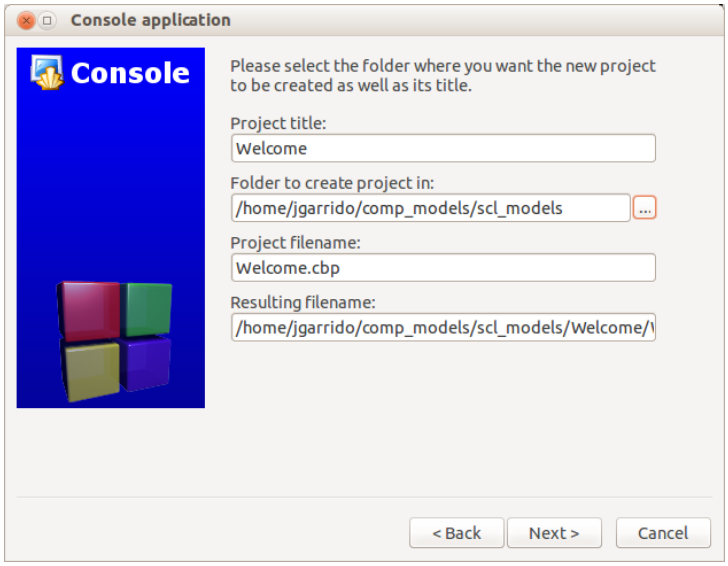

Figure 4: Title and location of project.

4. Select GNU GCC Compiler and click the Finish button.

5. Activate the left pane of the screen (Management), click the Projects tab. Remove the C source file 'main.c' by right-clicking on it. The file appears on the editor pane. Now you can edit this C source file.

6. A new source file can be created by selecting File menu, then New, and Empty file. Click Yes to add this empty to the project. A new dialog window appears, type the name of the file with its `scl` extension. Now you can start editing this source file and when finished, save the file.

7. If one or more existing source files are to be included in the project, select the Project menu on the top bar, or right-click on the project name. Select Add files. Select the directory of the source to add to the project and select the SPSCL source file(s). Click Open, see Figure 5.

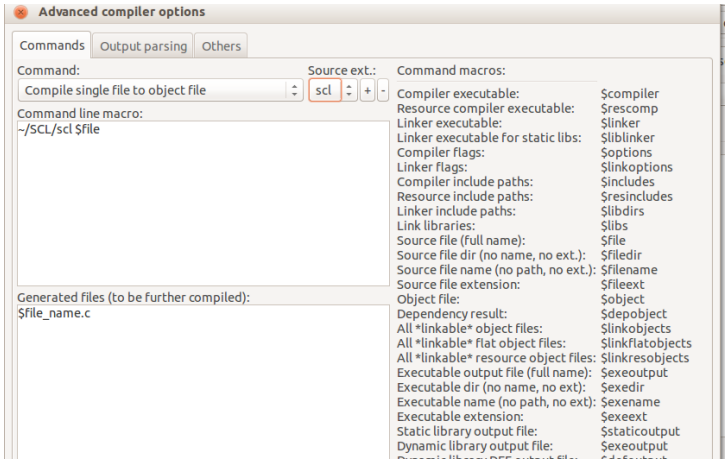

Figure 5: Adding an existing SPSCL source file to a project.

8. The source files are now under 'Others' and under the project name. Double-click on the desired source file to edit it further. The file now appears on the edit area of the screen

9. Right-click on the current project name, or activate the Project menu in the top bar, and select 'Build options'.

   (a) On the tab 'Compiler settings', check 'Enable all compiler warnings'.

   (b) This is an optional step, for small programs may not be required. On the tab 'Linker settings', click the 'Add' button to add a library to the project. Repeat to add all necessary object files and libraries. For example, the `m` (standard math library). This is shown in Figure 6.

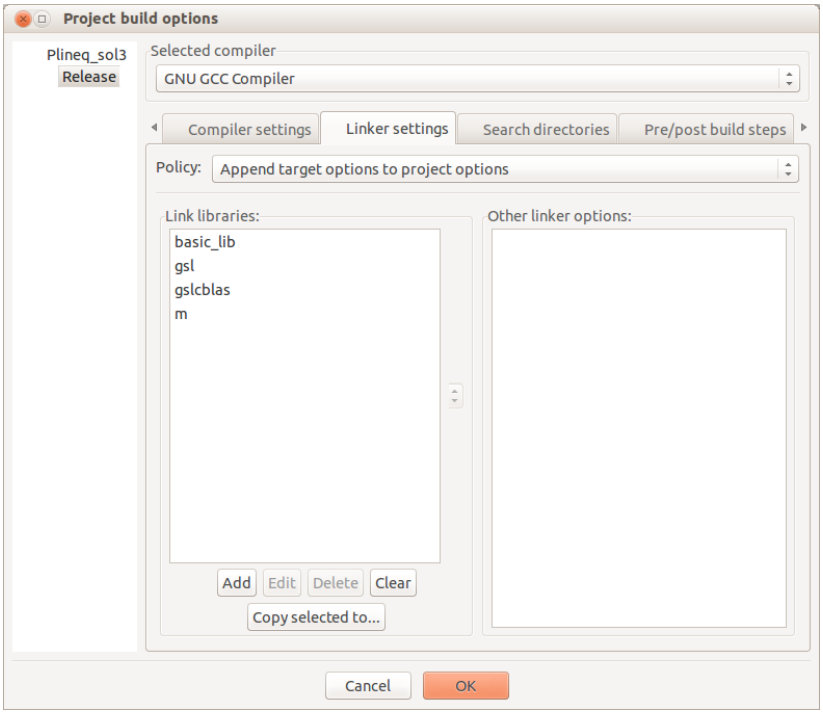

Figure 6: Linker settings.

(c) On the tab 'Search directories' and the tab 'Compiler', add the search directory for header files required by the source program while compiling. In Figure 7, the header file required by the program is located in the folder `~/SPSCL`.

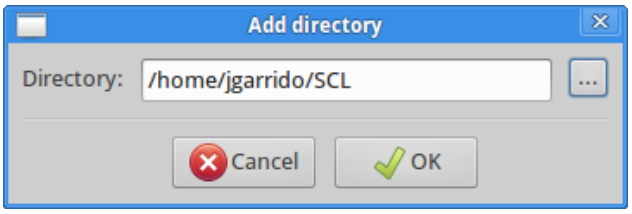

Figure 7: Adding a compiler search directory.

(d) On the tab 'Search directories' and the tab 'Linker', add the search directory of the libraries if needed. On Linux Ubuntu, the external libraries are always stored in standard system directories.

10. Build the project, which translates the SPSCL source file, compiles, and links the files in the project. On the top bar, select the Build menu and select the Build option. The Build log appears in the lower pane of the Codeblocks screen.

11. To execute the program, select the Run option in the Build menu. A new screen appears with the results of the execution, as shown in Figure 8. After the program terminates execution, press the Enter key.

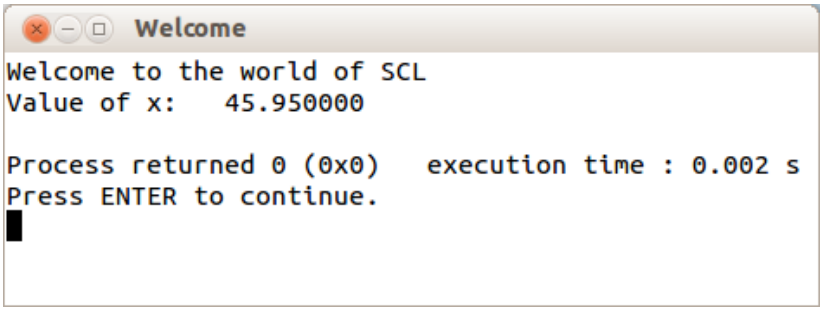

Figure 8: Executing the project.

12. On the top bar, activate the File menu and select Close project. Because the project was created in the directory `~/scl/scl_progs`, Codeblocks creates several new directories and the executable file in located in the directory:

    `~/scl/scl_progs/Welcome/bin/Debug`