

## Chapter 2

# Downloading, Configuring, Installing, and Starting with SystemC

**Abstract** Having introduced the SystemC [1] library in the previous chapter, we discuss in detail how it might be downloaded, compiled, and installed on one's computer. As the SystemC [1] library was originally developed and still very widely used on computers running the Linux and Linux-like operating systems, the emphasis is on installing on Linux machines. It can also be installed and run on computers running Microsoft Windows operating system as well, though the steps are much more complicated.

### 2.1 Downloading and Installing SystemC

SystemC [1] is an ANSI C/C++ extension library originally developed and runs best on computers using Linux/Unix-like operating systems. The source file tarballs may be downloaded from the official Open SystemC Initiative (OSCI) Web site (<http://www.accellera.org>), decompressed, configured, compiled (with the familiar gcc C/C++ compiler or the Solaris compiler), installed, and verified for correct compilation and installation. It can be installed on a computer running the Windows operating system, following one of two methods. The straightforward method is to create a Linux/Unix-like environment by installing the popular Cygwin/MingW packages and then installing/running SystemC through it. The slightly more complicated way is to compile and run SystemC directly with Microsoft Visual C++. All code examples in this book have been compiled and tested on a machine running Red Hat Enterprise Linux 4.0 (GCC compiler version 3.2.6) and a computer running Red Hat Fedora 14 and Fedora 15 (GCC compiler version 4.5.1). All design examples in this book have been compiled and tested to run both under older stable SystemC 2.2.0 release and under the brand new stable SystemC 2.3 release. A very important distinction between SystemC 2.2.0 and 2.3.0 is that the latter runs as a shared library, and so either uses the runtime environment variable `LD_LIBRARY_PATH` or adds the linker flags at compile time, to the compile script. In addition, if security-enabled Linux (SELinux) is enabled,

it can be disabled (temporarily using ‘setenforce’ or permanently using ‘execstack’ Linux system commands) easily. These details will be enumerated when design examples are discussed.

Compilation/installation of SystemC [1] on any computer running the Linux/Unix-like operating system is achieved by executing the familiar sequence of commands (*configure*, *gmake*, *gmake install*, and *gmake check*) from the command line. The following is an excerpt from the INSTALL file in the SystemC [1] source directory. Each design example in this book has been verified to run with the recent stable SystemC release 2.3.0 and previous stable release 2.2.0.

#### INSTALL NOTES FOR SystemC release 2.3.0

##### Contents:

1. Installation notes for Unix
2. Installation notes for Windows.

##### 1. Installation notes for Unix

##### System Requirements

SystemC can be installed on the following UNIX or UNIX-like platforms:

- 32-bit Linux (x86) (Red Hat Enterprise Linux 4, 5, 6; Fedora 14, 15; Debian 5.0; Ubuntu 10.04LTS, 12.04LTS) with GNU C++ compiler versions gcc-3.3.2 through gcc-4.7.0
- 32-bit Linux (x86) (Debian, Ubuntu 12.04LTS) with Clang C++ compiler version clang-2.9 through clang-3.1
- 64-bit Linux (x86\_64) (Red Hat Enterprise Linux 4, 5, 6; Fedora 17; Debian 5.0; Ubuntu 10.04LTS, 12.04LTS) with GNU C++ compiler versions gcc-3.4.5 through gcc-4.7.0
- 64-bit Linux (x86\_64) (Ubuntu 12.04LTS) with Clang C++ compiler.

#### INSTALL NOTES FOR SystemC release 2.2.0

##### Contents:

1. Installation notes for Unix
2. Installation notes for Windows.

##### 1. Installation notes for Unix

##### System Requirements

SystemC can be installed on the following UNIX platforms:

- Sun Solaris 2.8 with GNU C++ compiler versions gcc-2.95.3 and gcc-3.2.3.

The ‘gcc/g++’ is a common C/C++ compiler suite on Linux or Unix-like operating systems. All examples have been compiled/tested with both g++ 4.5.1 and 4.7.1, and SystemC versions 2.2.0 and 2.3.0, running on Fedora 14, 15 and Red Hat Enterprise Linux 4. There is a difference between how SystemC code is executed under SystemC 2.2.0 and SystemC 2.3.0. Note that any SystemC [1] code may be compiled easily with a command as:

```
g++ -I -I<absolute or relative path to SystemC directory>/include -L -L<absolute or relative path to SystemC directory>/lib-linux -o <name of executable> <executable source file name>.cc -lsystemc -lm.
```

Note that it is very common to name the executable file as ‘sim’. There is no restriction however, and it may be named anything. A Makefile may be used.

Executing under SystemC 2.2.0 is easy from command line, just by typing in `./<executable file name>` at the shell command prompt. Just like any other C/C++ program, any number of command line arguments may be passed into the executable.

Execution under SystemC 2.3.0 is different, because now SystemC is run as a shared library, and the user has to set and use the `LD_LIBRARY_PATH` environment variable. The following steps show the details, for a simple example named `test.cc`. We start with compiling:

```
g++ -I -I< absolute or relative path to SystemC directory>/include -L -L<absolute or relative path to SystemC directory>/lib-linux -o sim test.cc -lsystemc -lm.
```

Setting the `LD_LIBRARY_PATH` environment variable(*bash* shell):

```
export LD_LIBRARY_PATH=<absolute or relative path to SystemC directory>/lib-linux.
```

On the ‘cshrc’ or ‘tcshrc’ shells, use the ‘setenv’ command (note differences with bash shell):

```
setenv LD_LIBRARY_PATH<absolute or relative path to SystemC directory>/lib-linux.
```

Now execute with `./<executable file name>`.

Sometimes, security-enhanced Linux (SELinux) is enabled by default. Obviously then, it has to be disabled temporarily for the SystemC 2.3.0 session. It may be permanently disabled as well, but is risky. Either way, system administrator or root access is required. Once the system administrator has logged in, SELinux may be disabled with the Linux system command ‘*setenforce*’—‘*setenforce 0*’ disables SELinux security temporarily, while ‘*setenforce 1*’ enables all SELinux security features. SELinux may be disabled permanently with the Linux system command ‘*execstack*’. All of this apply to SystemC 2.3.0 only.

## Reference

1. IEEE Standards Board, IEEE Standards Association Standards Board IEEE-SA — IEEE Get Program. IEEE, 2011 *IEEE Standard 1666 Open SystemC Language Reference Manual (LRM)* <http://standards.ieee.org/getieee/1666/download/1666-2011.pdf>

<http://www.springer.com/978-3-319-01146-2>

SystemC and SystemC-AMS in Practice

SystemC 2.3, 2.2 and SystemC-AMS 1.0

Banerjee, A.; Sur, B.

2014, IX, 460 p. 224 illus., 31 illus. in color., Hardcover

ISBN: 978-3-319-01146-2