SystemCシミュレータを使ってみよう

塚田雄一

ここでは，Open SystemC Initiative（OSCI）が提供しているSystemCシミュレータなどをダウンロードし，SystemCの動作環境を整える方法について説明する．　（編集部）

SystemCは，Open SystemC Initiative（OSCI）によって規格化されたハードウェア・モデリング言語です．システムのモデリングのために必要な，ハードウェアのタイミングや並列動作，イベントへの応答などの構文を定義しています．OSCIは，SystemCのC++クラス・ライブラリ（シミュレーション・カーネル）を無償で提供しています．これらを入手し，設計対象のモデルと合わせてC++コンパイラでコンパイル（make）することにより，検証用の実行ファイル（SystemCシミュレータ）を作成できます．

SystemCシミュレータを動作させるには，GNUやVisual Studio（Visual C++）といったC++の動作環境が必要です．逆に言うと，C++の環境さえあれば，だれでも無料でシミュレーション環境を構築できます．本稿では，Linux（ここではRed Hat Linux 9で動作を確認）とGNUコンパイラ（同gcc 3.2.2）を使って，SystemCモデリング環境を構築する方法を説明します．

● Webサイトから必要なファイルを入手する

まず，OSCIのWebサイト（URLは「http://www.systemc.org/」，**図1**）から「ダウンロード」をクリックし，systemc-2.0.1.tgzとregtest-2.0.1.tgzをダウンロードします．初めてこのサイトからダウンロードする際には，アカウントを登録する必要があります．

● 解凍とインストールを実行する

ダウンロードしたファイルを解凍し，インストールを行います．このインストールについては，解凍したファイルの中の/systemc-2.0.1/INSTALLが参考になります．

1）ファイル解凍

ターミナルを開き，ダウンロードしたファイルを格納したディレクトリに移動して，以下のように解凍を行います．

```
$ tar -xvzf systemc-2.0.1.tgz
$ tar -xvzf regtests-2.0.1.tgz
```

2）動作環境設定

ターミナルにおいて，CXX環境変数の設定を行います．

```
$ export CXX=g++
```

3）Makefileの作成

ターミナルにおいて，作業ディレクトリ（objdir）を作成します．作業ディレクトリは，解凍したディレクトリsystemc-2.0.1の中に作成します．作業ディレクトリにおいて“../configure ”を実行し，makeに必要なMakefileを作成します．

```
$ cd systemc-2.0.1
$ mkdir objdir
$ cd objdir
$ ../configure
```

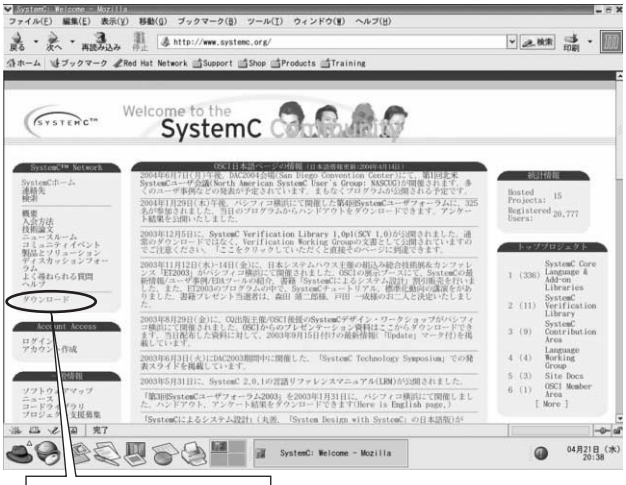

図1　OSCIのWebサイト
URLは「http://www.systemc.org/」．

4）インストール作業

ターミナルにおいてGNUのmake（gmake）を実行します．実行には数十分かかりますが，実行が完了すればインストール作業は完了です．systemc-2.0.1ディレクトリの中にinclude，lib-linuxディレクトリが作成されていれば，正常にインストールされています．作成されていない場合は，再度インストール作業を行ってください．

```
$ gmake
$ gmake debug
$ gmake install
```

● サンプルを動作させてみる

例として，今回はpipe[注1]を動作させることにしました．systemc-2.0.1ディレクトリの中のexamplesの下のpipeディレクトリに移動し，gmakeを行います．run.xというファイルが作成されていれば正常に動作しています．

```
$ cd examples/systemc/pipe
$ gmake -f Makefile.linux
```

1）デバッグ実行

GNUのデバッガ（gdb）を起動し，r（実行）コマンドを実行すると，シミュレーション結果（**図2**）が表示されます．

```
$ gdb run.x
(gdb) r
```

図2　gdbの実行
GNUのデバッガ（gdb）でpipeを実行したところ．表示される内容は，動作環境によって多少異なる．

2）ソース・レベル・デバッグの実行

ソース・レベル・デバッグを行うためには，デバッグ・オプションを設定してコンパイルmakeを行う必要があります．まず，テキスト・エディタ（emacsなど）でmakefile.linuxファイルを開きます．

```
(gdb) quit
$ emacs Makefile.linux
```

「CFLAGS=$(OPT) $(OTHER)」をコメント・アウトし（行頭に#を付ける），「CFLAGS=$(DEBUG) $(OTHER)」のコメントを解除してください（行頭の#を削除する）．

そして，再度コンパイルを行うことにより，デバッグ可能なサンプル・ファイルrun.xが作成されます．

```
$ gmake -f Makefile.linux
```

それではいよいよ，ソース・レベル・デバッグを実行します．先ほどと同様にgdbを起動します．ファイル・リストの表示（listコマンド），ブレーク（bコマンド），実行（rコマンド），ステップ実行（sコマンド）などを行うことができます．

```
$ gdb run.x
(gdb) list stage1.cpp:36
(gdb) b［行番号］
(gdb) r
(gdb) s
```

SystemC動作環境は無償で入手可能（フリー・ソフトウェア）ですが，より作業効率を上げたい場合は，ツール・ベンダなどが販売しているツールを利用するとよいでしょう[注2]．

つかだ・ゆういち
キャッツ（株）

＜筆者プロフィール＞
塚田雄一．1989年，キャッツ（URLは「http://www.zipc.com/」）に入社．以来15年間，OEM製品，自社開発製品（EMUSE）の開発に従事．現在はSystemC関連のツール「XModelink」を担当している．

注1：pipeは三つのモジュールをパイプラインでつないだサンプルである．
注2：例えばキャッツは，SystemC関連ツールとして「XModelink SystemCデバッガ」を販売している．Windows（VC++）用の体験版は，本誌の2004年4月号の付属CD-ROMに収録されている．