EECS 222: Embedded System Modeling
Spring 2019

Assignment 2

Posted: January 15, 2019
Due: January 23, 2019 at 6pm

Topic: Introduction to SpecC Compiler and Simulator

1. Setup:

We will use the same Linux account and the same remote servers as for Assignment 1.

For this and the following assignments, however, we will also use special software, namely the SpecC tools, which have been installed on the servers for this course. After logging into a new shell terminal, you will need to configure your Linux environment so that the SpecC tools and libraries are found in your search path and can be run. This configuration depends on the shell you are using, which you can identify with the following command:

```
echo $SHELL
```

If you use the **csh** or **tcsh** shell, then setup the SpecC environment as follows:

```
source /opt/sce/bin/setup.csh
```

On the other hand, if you use the **sh** or **bash** shell, then setup the SpecC environment like this (note the dot command at the beginning!):

```
. /opt/sce/bin/setup.sh
```

The setup script will configure your search path and other environment variables to the latest version of the SpecC compiler, simulator, and other tools.

2. Task A: Simple SpecC Examples

As an initial step to get familiar with the SpecC compiler and simulator, you should examine, compile, and run some of the provided simple examples.

The following commands show a typical session for this task:

```
mkdir simple_tests
cd simple_tests
cp /opt/sce/examples/simple/* ./
ls
vi HelloWorld.sc
scc HelloWorld -vv
./HelloWorld
ls
man scc
vi BitVectors.sc
scc BitVectors –vv
./BitVectors
...
```

Inspect, compile, and simulate also the other examples!

3. Task B: Producer-Consumer Example

As deliverable for this assignment, write and simulate a SpecC program of a simple Producer-Consumer example.

Your program should contain two parallel behaviors named **`Prod`** and **`Cons`** that communicate via an interconnecting channel **`C`**. The producer should send the message **`"Beans and Potatoes"`** to the consumer. This communication should be performed character by character (a single byte at a time should be sent/received through the channel). While sending and receiving the characters, the consumer and producer behaviors should print each sent/received character to the screen. After the entire message is communicated, both consumer and producer should terminate so that the simulation cleanly completes.

Your SpecC program output should look like this:

```
Main starts.
Producer starts.
Producer sends 'B'.
Consumer starts.
Consumer received 'B'.
Producer sends 'e'.
Consumer received 'e'.
```

```
Producer sends 'a'.
Consumer received 'a'.
Producer sends 'n'.
Consumer received 'n'.
Producer sends 's'.
Consumer received 's'.
[...]
Producer sends 's'.
Consumer received 's'.
Consumer done.
Producer done.
Main done.
```

For this task, create a new directory named **hw2** and work inside it. Name your SpecC source code file **ProdCons.sc**. Type the file name exactly as shown, since you won't be able to submit it otherwise.

Hint: To create your SpecC model, review the channel communication presented in Lecture 4. Your program should be similar to the example discussed there, with only a few modifications and some additional statements (e.g. to print the lines shown above).

To demonstrate that your program compiles (without warnings!) and simulates as expected, compile and run it. Redirect the simulation output into a file named **ProdCons.log**. Again, use exactly this filename, otherwise you cannot submit it.

3. Submission:

For this assignment, turn in the following deliverables:

```
ProdCons.sc
ProdCons.log
```

To submit these files, change into the parent directory of your **hw2** directory and run the **~eecs222/bin/turnin.sh** script. This command will locate the current assignment deliverables and allow you to submit them, as follows:

```
doemer@bondi.eecs.uci.edu: ~eecs222/bin/turnin.sh
================================================
EECS222 Winter 2019:
Assignment "hw2" submission for doemer
Due date: Wed Jan 23 18:00:00 2019
** Looking for files:
**   ProdCons.sc
**   ProdCons.log
================================================
```

```
* Please confirm the following:                        *
* "Following the Academic Honesty Policy at UCI,       *
* I submit my own original work."                      *
* Please type YES to confirm. y
=====================================================
Submit ProdCons.log [yes, no]? y
  File ProdCons.log has been submitted
Submit ProdCons.sc [yes, no]? y
  File ProdCons.sc has been submitted
=====================================================
  Summary:
=====================================================
Submitted on Mon Jan 15 08:28:14 2019
You just submitted file(s):
  ProdCons.log
  ProdCons.sc
```

Note that you can use this turnin-script to submit your work at any time before the deadline, *but not after!* Since you can submit as many times as you want (newer submissions will simply overwrite older ones), it is highly recommended to submit early and even incomplete work, in order to avoid missing the deadline.

The deadline is hard. Late submissions will not be considered!

As before, to double-check that your submitted files have been received, you can run the **`~eecs222/bin/listfiles.py`** script.

For any technical questions, please use the course message board.

--
Rainer Dömer (EH3217, x4-9007, doemer@uci.edu)