

Assignment 1 COMP[39]151 11s2

Dancing with the Shakes

PRE-RELEASE VERION 2 — FULL VERSION TO COME

Kai Engelhardt

Revision: 1.3 of Date: 2011/08/22 00:12:43

This assignment is worth 15 marks and due before the logical end of week 8, that is, **Wednesday September 21st, 12:59:59** local time Sydney.

Problem Statement

You have recently been hired by a start-up company called *Rattle & Shake System Solutions* (AKA RS³) specialising in concurrent soft- and hardware solutions for the persioners' entertainment industry. Your first job is to deliver an efficient scheduling system for the new season of “Dancing with the Shakes”, channel 12's award-winning reality show for the elderly.

Ratings for the show have been shown to be strongly correlated to the popularity of the performers getting the most air time.

Channel 12 has hired a number of professional dancers to partner with the real stars—a handful of carefully selected aged persons who happen to be somewhat capable of dancing. There is a real audience as well as millions glued to the TV screens if what you do works well. Audience members (local and remote alike) can use a smart phone app to request a certain dancer to take to the stage. At any given time there should be two dancers on stage, dancing to a song. After their song has ended, they both vacate the stage for the next two, necessarily different dancers. Since this is Channel 12 there is the rule that at least one of the two performers has to be from the aged persons group. The complete set of rules is:

1. At any given time at most one show by precisely two dancers of which at least one is aged is happening on stage.
2. Nobody dances for two songs in a row.
3. Each audience member loops forever through four phases: vegetate (this can lead to nodding off terminally), request any one of the dancers (independently of what other audience members request), watch said dancer dance on stage/TV, witness said dancer leave the stage for some well-earned rest.
4. If audience member m requests dancer d while no other audience member is requesting or watching a conflicting dancer, then m should see d on stage/TV within a bounded number of his own steps (in the requesting phase).

5. If audience member m requests dancer d then m must eventually watch d dance.
6. Dancers never die. Audience members only die while being vegetative.

Task

Solve the problem stated above by designing a concurrent algorithm. Implement it in C with pthreads so that it can be tested well. Encode the core algorithm and the relevant rules in Promela so that you can verify that the rules hold for sufficiently small numbers of performers and audience members. Prove your algorithm correct for any number of professional dancers, aged dancers, and audience sizes using pencil and paper methods. Write a compelling report on all that you've done.

Help Channel 12 with C + pthreads

Design and implement a simulation of your algorithmic solution. It should take a few command line arguments to determine the number p of professional dancers, u of aged dancers, n of audience members, and k of rounds each audience member should do thorough its four phases. Use $k = 0$ as a special case to loop indefinitely. There should a thread for each dancer and each audience member.

Save Channel 12's Ratings with Promela

Model the core of your scheduling algorithm and the relevant rules in Promela and attempt to verify that the rules are indeed satisfied by your model. It suffices to model the $k = 0$ case.

Document Your Exploits for the Archives of RS³

First of all, make it worth reading. Begin with a short summary of what you have achieved. Then elaborate on your achievements, e.g., by explaining how your algorithm works, what limitations you have encountered and implemented, what the time and communication complexities are, etc.

Do not mention Shawn.

Deliverables

`scs.c` is your C+pthread solution.

`scs.pml` is your Promela model, presumably with LTL formula definitions for the rules to be verified.

`scs.tex` is a \LaTeX document with your names or student numbers mentioned in the `\author` command. It contains your report.

Submission Instructions

The `give` command to be run is:

```
% 3151
% give cs3151 ass1 scs.c scs.pml scs.tex
```

The command above submits the bare minimum. Should you feel the need to include more files, e.g., for vector diagrams or include files, just list them as well.

Disclaimer

1. This assignment does not mandate field research.
2. All characters and events in this assignment specification—even those based on real people—are entirely fictional.