GPars - Groovy Parallel Systems

Jon Kerridge

Version 1.0, 2015-10-29

Table of Contents

Deadlock: An Introduction

Deadlock occurs whenever a network of processes gets into a state where none of the processes is able to continue execution.

A similar and related problem is that of **livelock**, which occurs when part of a process network operates in such a manner as to exclude some of the processes from execution, while others are able to continue. A first simple example, based upon the producer – consumer pattern, already discussed, demonstrates the ease with which a deadlocked system can be created.

Deadlocking Producer and Consumer

'Listing 7-1' gives the coding for a process **BadP**. The process has two channels (2, 3). Its run-method initially prints a starting message (5) after which it enters a loop (6). A message indicating the process is about to write to outChannel (7) is printed and then the output takes place (8). The same action is then undertaken for a read method on inChannel (9, 10). A message indicating that the end of the loop has been reached is printed (11) and the process loops back to (6).

Want to read more of this chapter? Download this chapter’s PDF here.