Solution of a Problem in Concurrent Programming Control

E. W. Dijkstra
Technological University, Eindhoven, The Netherlands

A number of mainly independent sequential-cyclic processes with restricted means of communication with each other can be made in such a way that at any moment one and only one of them is engaged in the "critical section" of its cycle.

Introduction

Given in this paper is a solution to a problem for which, to the knowledge of the author, has been an open question since at least 1962, irrespective of the solvability. The paper consists of three parts: the problem, the solution, and the proof. Although the setting of the problem might seem somewhat academic at first, the author trusts that anyone familiar with the logical problems that arise in computer coupling will appreciate the significance of the fact that this problem indeed can be solved.

The Problem

To begin, consider N computers, each engaged in a process which, for our aims, can be regarded as cyclic. In each of the cycles a so-called "critical section" occurs and the computers have to be programmed in such a way that at any moment only one of these N cyclic processes is in its critical section. In order to effectuate this mutual exclusion of critical-section execution the computers can communicate with each other via a common store. Writing a word into or nondestructively reading a word from this store are undividable operations; i.e., when two or more computers come simultaneously with the same common location, the computers have to be programmed in such a way that at any moment only one of them will enter its critical section first is postponed until eternity. In other words, constructions in which the computers have to be programmed in such a way that at any moment only one of them is engaged in the "critical section" of its cycle.

"After you"-blocking is still possible, although improbable, are not to be regarded as valid solutions.

The Solution

The common store consists of:

"Boolean array b, c[1..N]; integer k"

The integer k will satisfy 1 ≤ k ≤ N, b[i] and c[i] will only be set by the ith computer; they will be inspected by the others. It is assumed that all computers are started well outside their critical sections with all Boolean arrays mentioned set to true; the starting value of k is immaterial.

The program for the ith computer (1 ≤ i ≤ N) is:

"integer j;
Li0: b[i] := false;
Li1: if k ≠ i then
Li2: begin c[i] := true;
Li3: if b[k] then k := i;
Li4: go to Li5
end
else
Li5: begin c[i] := false;
for j := 1 step 1 until N do
   if j ≠ i and not c[j] then go to Li1
end;
critical section;
end;
b[i] := true; b[k] := true;
remaining of the cycle in which stopping is allowed;
go to Li0"

The Proof

We start by observing that the solution is safe in the sense that no two computers can be in their critical section simultaneously. For the only way to enter its critical section is the performance of the compound statement Li4 without jumping back to Li1, i.e., finding all other

computer can only request one one-way message at a time. And only this will make the reader realize to what extent this problem is far from trivial.