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. And only this will make the reader realize to what extent this problem is far from trivial.

The Solution

The common store consists of:

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

The integer k will satisfy $1 \leq k \leq N$, $b[i]$ and $c[i]$ will only be set by the $i$th 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 $i$th computer ($1 \leq i \leq N$) is:

```
integer j;
begin
    if j # i then k := i;
    if k # i then c[i] := true;
    for j := 1 step 1 until N
        if c[j] then begin
            if $b[k]$ then k := j;
            if $b[j]$ then j := k;
            if j = k then go to Lil;
        end;
    if k have been performed, c[i] := true after having set its own e to k does not change any more, the kth computer will wait until eternity. In other words, constructions in which "After you"-blocking is still possible, although improbable, are not to be regarded as valid solutions.

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 its critical section. In order to effectuate this mutual exclusion of critical-section execution the computers can try to communicate (either for reading or for writing) simultaneously with the same common location, i.e., finding all other computers and will not change its value for the time being, though one or mnore of them will find in their e true, as they will find k # i. And this, the author believes, completes the proof.

The Proof

We start by observing that the solution is safe in the sense that no two computers can be in their critical section simultaneouly. For the only way to enter its critical section is the performance of the compound statement "After you"-"After you"-blocking can occur; i.e., when two or more computers are engaged in the critical section, it must be impossible to devise for them a feeling for the tricky consequences of the fact that each one of them is engaged in the "critical section" of its cycle.

We beg the challenged reader to stop here for a while until eternity. In other words, constructions in which "After you"-blocking is still possible, although improbable, are not to be regarded as valid solutions.