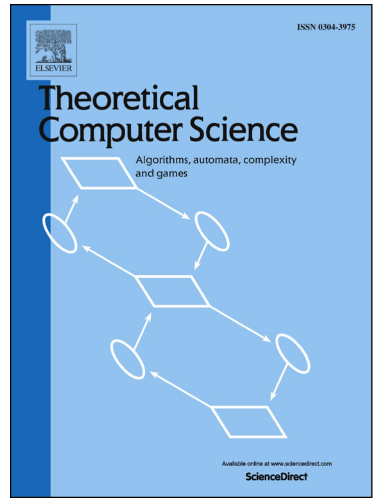
Accepted Manuscript



Two-process synchronization

Uri Abraham, Gal Amram

PII: S0304-3975(16)30050-0
DOI: http://dx.doi.org/10.1016/j.tcs.2016.04.011
Reference: TCS 10720

To appear in: *Theoretical Computer Science*

Received date: 25 March 2014
Revised date: 24 October 2015
Accepted date: 7 April 2016

Please cite this article in press as: U. Abraham, G. Amram, Two-process synchronization, *Theoret. Comput. Sci.* (2016), http://dx.doi.org/10.1016/j.tcs.2016.04.011

Two-Process Synchronization☆

Uri Abraham, Gal Amram*

Ben-Gurion University of the Negev, Beer-Sheva, Israel

Abstract

Aguilera, Gafni and Lamport introduced and solved the mailbox problem in [3], which can be used to speed-up interactions between a processor and a device. They provide a mailbox algorithm with "flag registers" of size 14 and space complexity $\Theta(n \log n)$, and they conjecture that a solution with logarithmic space complexity exists. We prove this conjecture by presenting such a mailbox algorithm. We prove that the space complexity of our mailbox algorithm is optimal. The size of its flag registers is reduced to 4, which matches a lower bound established by Aguilera et al. In the same paper, Aguilera et al. also pointed out that the mailbox problem can be solved by means of a more general problem: the signaling problem. They presented a non-blocking solution to the signaling problem, and asked whether a wait-free solution exists. We solve this question with a bounded wait-free signaling algorithm, and we prove its correctness. A related problem is the N-buffer problem introduced by Lamport. Aguilera et al. employ their mailbox algorithm and provide a first solution to the N-buffer problem with flag registers of constant size. The drawback of their solution is its space complexity which is unbounded. We show how our signaling algorithm can be used to solve the N-buffer problem with constant-size flags and optimal space complexity.

Keywords: distributed algorithms, shared memory, synchronization, linearizability

1. Introduction

The Mailbox Problem is a theoretical synchronization problem that arises from analyzing the situation in which a processor must cater to occasional requests from some device. The problem, as presented (and solved) in [3] (see also [2]) requires the implementation of three operations: deliver, check, and remove. The device executes a deliver operation whenever it wants to get the processor's attention, and the processor executes from time to time check operations to find out if there are any unhandled device requests. After receiving a positive answer for its check operation the processor executes a remove operation to find-out the nature of the request and to clear the interrupt controller. In a serial execution, it is required that a check operation C returns a positive answer if and only if the number of deliver occurrences that precede C is strictly greater than the number of remove operations executed before C. The mailbox object is specified by the following atomic code, in which the two processes share a queue of messages Q:

deliver(*letter*)	remove	check
1. add *letter* to Q D_num := D_num + 1	1. remove a letter from Q R_num := R_num + 1	1. if D_num > R_num return true else return false

The mailbox problem is to design a deliver/check/remove algorithm in which the check operation is as efficient as possible, namely that it employs bounded registers (called "flags") that are as small as possible

☆This paper combines and expends results that appeared in two conference articles: "On the Signaling Problem" (ICDCN 2014) [5] and "On the Mailbox Problem"(OPODIS 2014) [1].

*Corresponding author

Email addresses: abraham@cs.bgu.ac.il (Uri Abraham), galamra@bgu.cs.ac.il (Gal Amram)