FRI-ONLINE-1-CCT1-01

IMPLEMENTATION OF THE *CSP* SEMANTICS OF INTER-PROCESS COMMUNICATIONS USING THE *C++11* STANDARD LIBRARY[1]

Assoc. Prof. Milen Loukantchevsky, PhD, IEEE Member, ACM Member
Department of Computer Systems & Technologies,
University of Ruse
Phone: 0877 303 850
E-mail: mil@ieee.org

Abstract: *The theory of Communicating Sequential Processes (CSP) is a basis for analysis and synthesis of concurrent computer systems. Its uniqueness is in its simultaneous manifestation as an abstract model, a parallel systems specification language and a parallel programming language. Over its long history, CSP has found numerous of successful implementations - from transputers and its OCCAM language to modern Go and Python languages. The constructive potential of this parallel computational model despite of that is far from being expended.*

From engineering point of view CSP defines an abstract parallel machine, CSP machine. It consists of two subsystems: a set of computing nodes and a communicating environment. in each of the computing nodes are executed one or more concurrent asynchronous processes. The communicating environment consists of set of channels. The processes grouped by pairs could communicate only by message passing over a communication channel.

The CSP semantics of inter-process communications suggest the simplest possible kind of channels – unidirectional, one-to-one (1:1), unbuffered, with direct naming. Such channels do not actually imply flow control but required a special kind of bilateral synchronization known as "rendezvous".

Whiles std::threads could be regarded near close to CSP::process semantics, in C++ standard library there is not type even distant from CSP::channel. Hence the main goal set: proposal of suitable implementation of the CSP semantics of inter-process communications with C++11 standard library means. as result a csp namespace is defined. It includes the type csp::chan, nearly strong implementation of CSP::channel semantics, with two methods csp::chan::send() and csp::chan::recv() corresponding to CSP communications commands ! and ?. The alternative command for non-deterministic selection is implemented by the method csp::alt() using multiple wait on std::condition_variable and randomized choice between true guards.

Keywords: *CSP, C++11, Multithreading, Concurrency, Non-deterministic Message Passing*

ASJC Codes: *1701, 1712*

REFERENCES

Abdallah, A., C. Jones, J. Sanders (Eds.). (2005). *Communicating Sequential Processes: the First 25 Years*. - Berlin: Springer-Verlag.

Bartlomiej, F. C++ Lambda Story. (2020). URL: https://leanpub.com/cpplambda (Accessed on 26 Sept. 2020).

C++ Reference: the condition_variable class. (2020). URL: https://en.cppreference.com/w/cpp/thread/condition_variable (Accessed on 26 Sept. 2020).

C++ Support in Clang. (2020). URL: http://clang.llvm.org/cxx_status.html#cxx17 (Accessed on 26 Sept. 2020).

Dijkstra, E. (1975). *Guarded Commands, nondeterminancy and formal derivation of programs*. // Communications of the ACM, Vol. 18, Num. 8, pp. 453-457.

Embarcadero RAD Studio Docwiki. (2020). URL: http://docwiki.embarcadero.com/RADStudio/Sydney/en (Accessed on 26 Sept. 2020).

Gregoire, M. (2020). *Professional C++*. 3rd Ed. - Indianapolis: John Wiley & Sons, Inc.

Hoare, C.A.R. (1978). Communicating Sequential Processes. // Communications of the ACM, Vol. 21, Num. 8, pp. 666-677.

[1] The paper is presented on 13 November 2020 and it is awarded with BEST PAPER prize. The paper is published in "Reports Awarded with BEST PAPER Cristal Prize"

Josuttis, N. (2019). *C++17 - the Complete Guide*. 1st Ed. URL: http://leanpub.com/cpp17 (Accessed on 26 Sept. 2020).

Josuttis, N. (2012). the *C++ Standard Library: a Tutorial and Reference*. 2nd Ed. - Boston: Pearson Education, Inc..

Josuttis, N., D. Vandevoorde. (2009). *C++ Templates: the Complete Guide*. - Boston: Pearson Education, Inc.

Loukantchevsky, M. (2013). *Modelirane na kvantovi izchisleniya v paralelna izpulnitelna sreda*. Ruse: Izdatelski tsentar na RU „A. Kanchev", ISBN 978-619-7071-25-2. (***Оригинално заглавие:*** Луканчевски, М. Моделиране на квантови изчисления в паралелна изпълнителна среда. Русе, Издателски център на РУ "А. Кънчев", 2013, ISBN 978-619-7071-25-2.)

May, D. (2009). the *XMOS XS1 Architecture*. – XMOS Ltd.

Watt, D. (2009). *Programming XC on XMOS Devices*. – XMOS Ltd.

Williams, A. (2012). *C++ Concurrency in Action: Practical Multithreading*. – Shelter Island: Manning Publications Co.