

Communicating Sequential Processes in Go*

Tutorial

Saverio Perugini¹ and David J. Wright²

¹Department of Computer Science

²Learning Teaching Center

University of Dayton

Dayton, Ohio 45469–2160

{saverio,dwight1}@udayton.edu

Project Site: <http://sites.udayton.edu/operatingsystems>

1 Tutorial Summary

We discuss the Communicating Sequential Processes (CSP) model of concurrency—a (re-)emerging concurrent programming model that is increasing in use, especially in the Go programming language.  Intended participants are instructors of undergraduate operating system (OS) courses. This tutorial is also of interest to instructors of concurrent programming and programming languages courses. We lead participants through two active-learning laboratory exercises for teaching a ‘concurrent programming and synchronization’ module of an undergraduate OS course using CSP in Go. The in-class laboratory plans that attendees can adopt involve programming classical problems of synchronization (e.g., producer-consumer and dining philosophers) using CSP in Go. Participants should have an understanding of concurrency and synchronization in a language such as Java or C and are encouraged to bring laptop computers to work through the lab activities. While the use of semaphores and similar synchronization constructs place an over-emphasis on the data that must be protected and the syntax and semantics of the level-low locking mechanisms which protect it, the CSP model facilitates the programmer’s active engage-

*Copyright is held by the author/owner.

¹<http://golang.org>

ment with the concurrent threads that are collaborating to solve a problem and the communication between them. Therefore, more broadly, we also anticipate fostering and facilitating a discussion of innovative approaches toward teaching the topic of concurrency and synchronization in an undergraduate OS course [1].

2 Schedule of Activities

- (10 minutes) Introduce attendees to the fundamentals of CSP and Go.
- (40 minutes) Work through two active-learning laboratory plans covering the CSP model in Go.
 - The Producer-Consumer problem
 - The Dining Philosophers problem
- (10 minutes) Discussion
 - Share our experience in teaching the OS course using this model.
 - How to use the active-learning labs in our Laboratory Manual.
 - Invite participation in our community of OS educators.
 - Questions and Answers.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant Numbers 1712406 and 1712404. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

- [1] S. Perugini and D.J. Wright. Developing a contemporary and innovative operating systems course. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE)*, page 1248, New York, NY, 2019. ACM Press. Conference Birds-of-a-Feather; DOI: <http://doi.acm.org/10.1145/3287324.3293734>.