Course Title: Multiprogramming for Parallel Processing

Course Code: 202-1-5401
Course Type: Elective Course
Credit: 4.0

Department: Faculty of Natural Sciences - Department of Computer Science
Semester: '87

Course Instructor: Dr. Ran Ateinger
Pre-requisite Courses: 202-1-2021-02 Computer Systems, 201-1-0201-02 Introduction to Logic and Set Theory

Course Objective: The course presents a method for planning and developing parallel algorithms and computer programs, directly from the specification (specification). This way, the correctness of the program (i.e., the program satisfies the properties specified in the specification) follows directly from the construction steps of the program.

The development process of parallel programs (multiprogramming) taught in this course is based on a simple theory, related to the fact that at each step in the development process, we know the requirements for proving its correctness. If we perform according to this method in each step of the development, we will get a verified program.

The basic idea of the theory (Fejes and van Gasteren) is the absence of interference (interference freeness) between processes in the program. According to this idea, if two processes communicate via a shared memory, the correctness of one process is not affected by the execution of another process, if we can show that the code of the other process does not affect the proof of the correctness of the first process.

The proof obtained in this method guarantees partial correctness (partial correctness): At the termination of the program, all variables will satisfy the properties specified in the specification. Additionally, we learn development methods to avoid critical situations due to deadlocks, such as reinitialization of shared variables and other.

The course is based on the book "On a Method of Multiprogramming" (Fejes and van Gasteren) and extends the theory of multiprogramming presented in the book with a wide variety of parallel algorithms, while learning parallel programming techniques to handle synchronization issues, deadlocks, and other problems.

The course is taught in Hebrew, and the materials are presented in Hebrew as well. The course is suitable for students with basic knowledge in computer science and mathematics.

Note: This course is an elective course, and it is recommended for students interested in computer science and parallel programming.
The following basic topics will be covered:

- A Computational Model
- Program Notation and its Semantics: Hoare triples and weakest-liberal-preconditions
- The Core of the Owicki/Gries Theory: annotating a multiprogram; postconditions
- Private Variables and Orthogonality
- System Invariants
- Mutual Exclusion
- Co-assertions and Strengthening the Annotation
- Synchronization and Total Deadlock
- Individual Progress and the Multibound

As time and progress permit, a modest selection of algorithms will be studied (both in class and home assignments), taken from the following collection (as it appears in the second part of the course textbook, starting at Chapter 13):

- The Safe Sluice
- Peterson’s Two-Component Mutual Exclusion Algorithm
- Handshake Protocols
- Phase Synchronization for Two Machines
- The Parallel Linear Search
- The Initialization Protocol
- The Non-Blocking Write Protocol
- Mutual Inclusion and Synchronous Communication
- A Simple Election Algorithm
- Peterson’s General Mutual Exclusion Algorithm
- Monitored Phase Synchronization
- Distributed Liberal Phase Synchronization
- Distributed Computation of a Spanning Tree
- Shmuel Safra’s Termination Detection Algorithm
- The Alternating Bit Protocol

The home assignments, midterm examination, and final project are all theoretical, in the sense that any amount of programming will be done “on paper”. The assignments and final project are to be prepared and submitted either individually or in pairs.

The final grade will be determined by two homework assignments (10% each), a must-pass midterm examination (20%), and a final project (60%).

The course is exclusively based on the monograph On a Method of Multiprogramming, by W.H.J. Feijen and A.J.M. van Gasteren.