

ACM SIGCSE Special Project Report

Dive into Systems

Suzanne J. Matthews, Tia Newhall and Kevin Webb

Project Summary

The purpose of our project is to develop and promote a free online textbook that covers introductory computer systems, architecture and parallel computing. The expense of modern textbooks often limits their availability to students and universities that can afford them. As computational thinking and programming increasingly becomes a desired skillset, cost-effective computing materials are needed to make computer science (CS) accessible to everyone. Furthermore, the ACM-IEEE CS2013 Curriculum [2] recommends at least 38 hours of System Fundamentals (SF) and Architecture & Organization (AR) coverage in the core computing program, along with an additional 15 hours of Parallel & Distributed Computing (PDC). The NSF/IEEE-TCPP Curriculum Initiative [3] curricular guidelines further specify expanded coverage of core systems and PDC topics. To help make these concepts accessible to everyone, we have developed a new, free, online textbook for teaching computer systems, architecture and PDC. To the best of our knowledge, this is the first textbook that incorporates all of these concepts and introduces them at an introductory-level, assuming only a CS1 background of the reader.

Our book fits a much needed niche in the computer science education community. To date, there are no free textbooks covering the breadth of topics that we cover in a single book. For example, while *Operating Systems in Three Easy Pieces* covers operating systems in depth, it understandably does not cover the breadth of topics, nor cover them at the introductory level, of our work. Another prominent example is *The Elements of Computer Systems* (Nand2Tetris). However, only the first six chapters are available for free, and only cover some basic concepts such as boolean logic and computer architecture. Furthermore, the assembly language presented in *The Elements of Computer Systems* is a custom assembly language (called Hack). Allen Downey has written a number of free textbooks, the most notable being *How to think like a computer scientist*. While there is a C version of the book, the coverage of C is mainly from a syntactical perspective. Notably, concepts such as memory allocation on the heap (through the use of functions like `malloc()`) and the use of debuggers is not discussed.

All the above mentioned textbooks are distributed in PDF. In contrast, *Dive into Systems* is available online, which makes it easy to read, search and update. *Dive into Systems* covers several topics, including introductory C, computer architecture, assembly programming (including chapters on IA32, x86_64, and ARM assembly), the memory hierarchy and caches, code optimization, operating systems, and parallel computing fundamentals. It covers these topics assuming only a CS1 background of the reader. The chapters are designed to be as independent of each other as possible, enabling an instructor to mix and match chapters and sections for easy use in a variety of courses.

We used the funds of the SIGCSE grant to recruit CS faculty from across the country to adopt *Dive into Systems* in a variety of systems courses. We offered each early adopter a \$100 stipend in exchange for piloting the book in their course and for completing an end-of-the-semester survey. We list the details of our 27 early adopters below. Note that many of our early adopters elected not to receive a stipend, so we ultimately paid out less funding to early adopters than we originally anticipated. Our project also attracted the attention of a well-known publisher, who will be producing a low-cost print edition of the book. As such, we will be returning the money associated with copy editing to the SIGCSE organization.

Our textbook is available at: <https://diveintosystems.org>

Project Status and Results

2019-2020 Early Adopters (note: several declined the stipend)

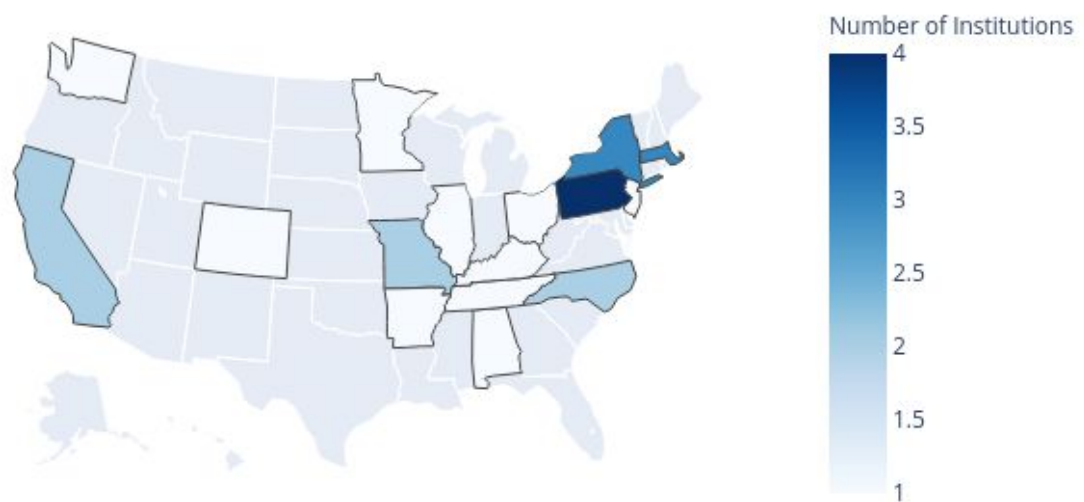
The following individuals piloted *Dive into Systems* as a textbook at their institutions during the 2019-2020 Academic Year:

1. John Barr (Ithaca College) - Computer Organization & Assembly Language (Comp 210)
2. Chris Branton (Drury University) - Computer Systems Concepts (CSCI 342)
3. Dick Brown (St. Olaf College) - Hardware Design (CSCI 241)
4. David Bunde (Knox College) - Introduction to Computing Systems (CS 214)
5. Bruce Char (Drexel University) - Systems Programming (CS 283)
6. Vasanta Chaganti (Swarthmore College) - Introduction to Computer Systems (CS 31)
7. Bryan Chin (U.C. San Diego) - Computer Organization and Systems Programming (CSE 30)
8. Stephen Carl (Sewanee: The University of the South) - Computer Systems and Organization (CSci 270)
9. John Dougherty (Haverford College) - Computer Organization (cs240)
10. John Foley (Smith College) - Operating Systems (CSC 262)
11. Elizabeth Johnson (Xavier University) - Programming in C
12. Alexander Kendrowitch (West Point) - Computer Organization (CS380)
13. Bill Kerney (Clovis Community College) - Assembly Programming (CSCI 45)
14. Deborah Knox (The College of New Jersey) - Computer Architecture (CSC 325)
15. Doug MacGregor (Western Colorado University) - Operating Systems/Architecture (CS 330)
16. Jeff Matocha (Ouachita Baptist University) - Computer Organization (CSCI 3093)
17. Keith Muller (U.C. San Diego) - Computer Organization and Systems Programming (CSE 30)
18. Crystal Peng (Park University) - Computer Architecture (CS 319)
19. Leo Porter (U.C. San Diego) - Introduction to Computer Architecture (CSE 141)
20. Lauren Provost (Simmons University) - Computer Architecture and Organization (CS 226)

21. Kathleen Riley (Bryn Mawr College) - Principles of Computer Organization (CMSC B240)
22. Roger Shore (High Point University) - Computer Systems (CSC-2410)
23. Tony Tong (Wheaton College, Norton MA) - Advanced Topics in Computer Science: Parallel and Distributed Computing (COMP 398)
24. Brian Toone (Samford University) - Computer Organization and Architecture (COSC 305)
25. David Toth (Centre College) - Systems Programming (CSC 280)
26. Bryce Wiedenbeck (Davidson College) - Computer Organization (CSC 250)
27. Richard Weiss (The Evergreen State College) - Computer Science Foundations: Computer Architecture (CSF)

The following map shows the geographical distribution of our early adopters:

Dive into Systems Early Adopters (By State)



Report of Spending

A total of 27 faculty (including 25 from external institutions) participated in our early adopter program. Of the 25 faculty, 12 faculty elected not to receive the stipend. Thus, we paid out 13 stipends totalling \$1,300.00 in stipend money to our early adopters.

As a secondary expense, we planned on spending money for copy editing (to aid in self-publishing). However, our book caught the attention of No Starch Press, and they are in the process of creating a low-cost print edition of our book. As such, we no longer needed the \$3,700.00 requested for copy-editing, and are returning it to the SIGCSE board.

Expense Category	Projected	Actual
Early Adopter Stipends	\$2,000.00	\$1,300.00
Copy-Editing	\$3,000.00	0.00
Total	\$5,000.00	\$1,300.00
Amount to Return		\$3,700.00

Summary

The 2019-2020 Early Adopter program was a huge success; faculty reported enjoying using the book, and most faculty expressed a desire to use the book again. The fact that we did not use the SIGCSE funds for copy-editing is not considered a failure; on the contrary, the fact that the book has attracted the attention of a well-known publisher speaks to the quality of the final product. We are excited to see where the project takes us next. Visit the book project at <https://diveintosystems.org/>.

References:

- [1] CSForAll. <https://www.csforall.org/>
- [2] Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science, The ACM-IEEE Joint Task Force on Computing Curricula, 2013, (<https://www.acm.org/education/curricula-recommendations>).
- [3] NSF/IEEE-TCPP Curriculum Initiative on Parallel and Distributed Computing - Core Topics for Undergraduates, Prasad et al., 2012, (<http://www.cs.gsu.edu/~tcpp/curriculum/>).