Dive into Systems: A Free, Online Textbook for Introducing Computer Systems

Suzanne J. Matthews
suzanne.matthews@westpoint.edu
U.S. Military Academy
West Point, NY, USA

Tia Newhall
newhall@cs.swarthmore.edu
Swarthmore College
Swarthmore, PA, USA

Kevin C. Webb
kwebb@cs.swarthmore.edu
Swarthmore College
Swarthmore, PA, USA

ABSTRACT

This paper presents our experiences, motivations, and goals for developing *Dive into Systems* [17], a new, free, online textbook that introduces computer systems, computer organization, and parallel computing. Our book's topic coverage is designed to give readers a gentle and broad introduction to these important topics. It teaches the fundamentals of computer systems and architecture, introduces skills for writing efficient programs, and provides necessary background to prepare students for advanced study in computer systems topics. Our book assumes only a CS1 background of the reader and is designed to be useful to a range of courses—as a primary textbook for courses that introduce computer systems topics or as an auxiliary textbook to provide systems background in other courses. Results of an evaluation from students and faculty at 18 institutions who used a beta release of our book show overwhelmingly strong support for its coverage of computer systems topics, its readability, and its availability. Chapters are reviewed and edited by external volunteers from the CS education community. Their feedback, as well as that of student and faculty users, is continuously incorporated into its online content. We anticipate releasing version 1.0 of the book in spring of 2021, and a release candidate is currently available at https://diveintosystems.org.

CCS CONCEPTS

• **Applied computing** → **Education**; *Publishing*; • **Social and professional topics** → *Student assessment*; • **Computing methodologies** → *Parallel computing methodologies.*

KEYWORDS

textbook, free, online, computer systems, parallel computing

ACM Reference Format:
Suzanne J. Matthews, Tia Newhall, and Kevin C. Webb. 2021. *Dive into Systems*: A Free, Online Textbook for Introducing Computer Systems. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21), March 13–20, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3408877.3432514

SIGCSE '21, March 13–20, 2021, Virtual Event, USA.

ACM ISBN 978-1-4503-8062-1/21/03.
https://doi.org/10.1145/3408877.3432514

1 INTRODUCTION

From 1977 to 2015, textbook prices rose 1,041%, and the average U.S. college student can now expect to pay more than $1,200 per year for textbooks and supplies [23]. The expense of modern textbooks often limits their availability to those who can afford them. As computational thinking and programming increasingly become desired skills, making CS truly accessible to every student requires that the CS education community produces free or low-cost materials at *all levels* of the curriculum.

Recent ACM and IEEE curricular guidelines highlight the importance of systems topics. The ACM-IEEE CS2013 Curriculum [1] 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 Guidelines [20] further specify expanded coverage of core systems and PDC topics.

To help make these important concepts available to everyone, we developed a free online textbook, *Dive into Systems*, for teaching computer systems, computer organization, and PDC. *Dive into Systems* introduces a broad range of computer systems and organization topics, with two overarching themes of explaining how a computer runs a program and accounting for the systems costs associated with program execution. It serves as a gentle introduction to topics in computer systems, including C programming, architecture fundamentals, operating systems, assembly language, and multi-threading. In Section 3 we discuss our book's topic coverage and main themes in more detail.

In addition to its availability, *Dive into Systems*'s online format makes it a "living" textbook that can be rapidly updated with new content in response to changes in the computing field and feedback from users, enabling our textbook to stay current and useful. To the best of our knowledge, *Dive into Systems* 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.

Although initially motivated to write *Dive into Systems* due to difficulties finding a textbook that was a good fit for our own introductory computer systems and organization courses, we also wanted to share our textbook widely and to write it in such a way that it would be useful to a wide range of courses. As such, it serves as a primary textbook for introductory-level systems courses or as an auxiliary text for introducing systems and systems programming topics. Further, it functions as background reading on systems topics for upper-level courses such as OS, compilers, parallel and distributed computing, and computer architecture, or as a C or assembly programming reference.

Each chapter of *Dive into Systems* was peer-reviewed by several volunteer experts in a related field. In the fall of 2019, we released a beta version of *Dive into Systems* to participants in an early adopter program, which we describe in detail in Section 4. One of the goals of our early adopter program was to discover common and novel uses for our textbook and how well it fits into existing courses. To date, early adopters have incorporated the book into a wide rage of courses across more than 18 institutions. The numerous faculty who have used our textbook report a high degree of satisfaction with the level of presentation, the breadth and depth of topic coverage, its readability, the easy adoptability of our textbook for their course, and the advantages of its free, online format. In Section 5, we present an evaluation by early faculty adopters on their use of our textbook in their courses and by their students.

In making *Dive into Systems* free, we hope that the textbook will make one small (but critical) step in breaking down barriers to learning computing and to making computer systems, computer organization, and parallel computing concepts available to all. Our results suggest that it succeeds in meeting the needs of a broad range of systems courses and increases the availability of high quality, free computer science resources for all students.

2 RELATED WORK

Free, online textbooks represent an attractive alternative to traditional textbooks, due to both their cost and availability [13, 16]. Their adoption has been well regarded by students in many other disciplines, including psychology [15], nutrition [9], and business [5].

Most of the effort behind free teaching resources in CS focuses on introductory-level material. To date, there are no other free textbooks covering the breadth of systems topics, at the same curricular level, that *Dive into Systems* covers in a single book. *Operating Systems: Three Easy Pieces* [6] covers operating systems in depth, but it's targeted to upper-division courses. The first six chapters of *The Elements of Computer Systems (Nand2Tetris)* [18] are available online for free, however it only covers some basic concepts such as boolean logic and computer architecture using a custom assembly language named Hack.

Allen Downey has authored a number of free textbooks, the most notable being the *How to Think Like a Computer Scientist* series. While there is a C/C++ version of the book [3], its coverage of C mainly focuses on the syntax of the language, whereas *Dive into Systems* also uses C to explore the underlying systems concepts. His systems book, *Think OS* [4] is currently incomplete and lacks figures.

The free online textbook *Computer Organization and Design Fundamentals* [12] overlaps with some of *Dive into Systems*'s conceptual systems content (e.g., binary, architecture, and assembly). However, it primarily concentrates on hardware and low-level machine characteristics, whereas *Dive into Systems* broadly covers other aspects of computer systems (e.g., C programming, OS, and parallel computing). Other efforts like *Operating Systems: From 0 to 1* [21] and *Computer Science from the Bottom Up* [14] show promise as free online references for topics similar to *Dive into Systems*, but they appear to be in earlier stages of development and their updates seem to be infrequent.

The format of online learning materials enables new forms of student interaction and interactive engagement [2, 8, 10]. Although *Dive into Systems* does not currently take advantage of these opportunities, we intend to augment it with interactive student exercises as part of future work.

Finally, the CS education community recognizes many high-quality traditional print textbooks that cover systems topics [7, 11, 19, 22, 24], but none are freely available to students. Aside from cost, *Dive into Systems* also offers many advantages compared to the existing books such as its breadth of topic coverage, introductory style, and integration of three real assembly languages.

3 BOOK DETAILS

Dive into Systems introduces computer systems and parallel computing assuming only a CS1 background of the reader. We designed it for use as a primary textbook for intermediate-level courses, focusing on introductory computer systems, computer organization, and parallel computing. It is also designed for use as an auxiliary textbook for upper-level courses, providing background in systems topics, C programming, and assembly programming.

Our textbook topics cut a vertical slice through the computer, with three main themes:

- **How a computer runs a program.** We show how a program expressed in a high-level programming language translates to a form that can be executed by computer hardware, how the computer architecture is implemented to execute programs, and the role of the OS in enabling and controlling program execution on the computer hardware.
- **How systems costs affect program performance.** We illustrate the effects of the memory hierarchy on a program's performance and how to design a program to more efficiently access its memory. We also describe methods for evaluating the performance of sequential and parallel programs.
- **How to leverage the power of parallel computers and parallel programming.** We focus on shared memory parallelism on multicore architectures, introducing the OS's thread abstraction, issues with threading, and pthreads and OpenMP programming. We also provide an overview of other types of parallel and distributed computing.

3.1 Topics

The set of topics we cover, and our foci of coverage, is guided by the three primary themes described above. Briefly, the book's major topics include the following:

C Programming. After first introducing computer systems, the next three chapters introduce the C programming language and C debugging tools (GDB and Valgrind). We choose C in part because it is the language of systems programmers, but also because it is less abstracted from the computer's machine language, making it easier for a programmer to understand the relationship between their C program code and the computer's execution of it. We continue to provide C programming examples throughout the book, including in discussions of code optimization, translating a high-level language to assembly, and the architecture's execution of program instructions.

We first introduce features of the C language in a chapter that compares C to Python as an example of a first language with which a reader may be familiar. In a second chapter we cover C in depth, with a large focus on pointers and dynamic memory allocation, and include some advanced C programming features. In a third chapter, we cover C debugging tools GDB and Valgrind.

A later chapter presents some C code optimizations typically performed by a compiler and discusses how to design C programs to run more efficiently using knowledge about compiler optimizations and the memory hierarchy.

Binary Representation. In an early chapter, we cover binary representation of C's signed and unsigned integer types, operations on binary data (addition, subtraction, bit-wise operations), and overflow. We briefly present representations for real numbers.

Architecture. Our chapter on computer architecture focuses on CPU design for executing program instructions. We start with the Von Neumann architecture and a brief history of modern computer design. We then build a simple single-core CPU starting from logic gates, to arithmetic, memory, and control circuits that are combined together to implement the CPU. We discuss the clock circuit and demonstrate how it drives the stages of a program's instruction execution by the CPU. The chapter concludes by introducing pipelining and characterizing some features of today's CPUs.

Memory Hierarchy. A separate chapter introduces the memory hierarchy and its effects on performance. We discuss storage media and devices at different levels of the hierarchy and cover single-core CPU caching in detail, including direct mapped and set associative cache organizations. We discuss cache coherency for multi-core CPUs in a later chapter.

Assembly Programming. Over several chapters, we present assembly language programming. Our focus is on the mapping between C code and assembly code. Our goal is to demonstrate how high-level language code ultimately executes as a sequence of CPU instructions. We currently cover three different assembly language ISAs (32-bit x86, 64-bit x86, and 64-bit ARM) in three separate chapters. Each chapter presents similar content with a consistent structure, including: basic arithmetic instructions, memory access and movement instructions for assignment and loading and storing registers, conditional instructions for if-else and loops, instructions for function call and return, and stack instructions to support functions and local variable allocation. Instructors can choose one of these chapters to teach a specific assembly language or adopt multiple chapters to compare different assembly languages.

Operating Systems. The OS chapter focuses on what an OS is, how it works, and the role it plays in program execution. The chapter emphasizes the OS kernel and the mechanism part of the OS. We cover processes and virtual memory in some depth, including context switching and multiprogramming, process creation and exit (fork-exec-wait-exit), and virtual memory with a single-level page table implementation. We additionally include an overview of interprocess communication, including pipes and sockets, shared memory, and signals with a focus on handling SIGCHLD signals.

Parallelism. We devote two chapters to parallel computing and parallel systems, and there is some additional coverage of parallelism in other chapters throughout out the book. The first chapter on parallel computing emphasizes shared memory parallelism on multi-core processors, including threads and synchronization with in-depth coverage of pthreads programming. Additionally, we introduce OpenMP and cache coherency in multi-core architectures. The final chapter of our book presents a look ahead to other types of parallel and distributed systems and computing. It introduces accelerators and CUDA, distributed memory and MPI, and cloud computing and MapReduce.

3.2 Example Uses

Our goal of making *Dive into Systems* widely-applicable to a large set of courses led us to design chapters to be as independent as possible so that instructors can pick individual chapters and sections to fit the needs of their particular courses. To illustrate a few ways in which the textbook can be used, the following are examples of how the authors use it in different courses at their institutions [1]:

- Introduction to Computer Systems: as primary text.
- Computer Organization: as primary text.
- Compilers: as supplemental for C, assembly, optimization.
- Operating Systems: as supplemental for C, OS intro, memory hierarchy, threads, distributed computing.
- Architecture: as supplemental for memory hierarchy, architecture.
- Networking: as supplemental for C, OS intro, memory hierarchy, threads, distributed computing.
- Databases: as supplemental for C, memory hierarchy, parallel and distributed computing.
- Parallel and Distributed Computing: as supplemental for C, threads, parallel and distributed computing.

4 EARLY ADOPTER PROGRAM

In summer of 2019, we launched the Early Adopter program to encourage the use of *Dive into Systems* at colleges and universities around the United States. Each early adopter was offered a stipend of $100 in exchange for piloting the book as a required text at their institution, completing a survey about their experiences, and distributing a survey link to their students. We primarily recruited early adopters through the SIGCSE and EduPar mailing lists.

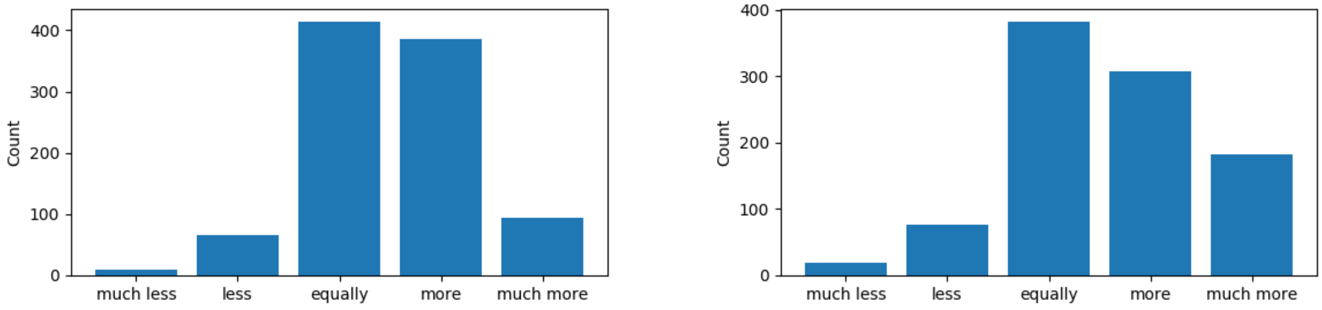
Including West Point and Swarthmore, a total of 20 institutions participated in the early adopter program. West Point was the IRB institution of record, and helped coordinate the IRB process at the participating institutions. Of the 20 candidate institutions, 18 completed the IRB review process. The two institutions who did not complete the IRB process were excluded from data collection.

Of the 18 institutions that participated in data collection, 15 (83.3%) were baccalaureate institutions. The remaining three institutions were a doctoral university, a masters-granting university, and a community college. The book was used as part of 23 course offerings at the 18 institutions over the 2019-2020 academic year. Of the 23 course offerings that used our book, 9 were used in the fall term and 14 were used in a winter or spring term.

5 EVALUATION

To evaluate the efficacy of *Dive into Systems* and study how the book was used, we created one survey for students and another for faculty. The goal of the student survey was to gain a student perspective on the usefulness of the textbook. The goal of the faculty

[1] The preface of our textbook provides more detail about the specific chapters we use for each of these. Additionally, our early adopters use our book in other ways in courses at their institutions.



(a) How helpful compared to other textbooks?

(b) Did the electronic nature of the book help you learn?

Figure 1: Summary of student survey data

survey was to study the contexts in which *Dive into Systems* was used and to see how well *Dive into Systems* fit as a text at various institutions. Both surveys asked participants to reflect on how the textbook compared to others that they have used.

5.1 Student Survey Results

We observed uneven response rates to the student surveys at the 18 different institutions that participated. Some faculty chose to incentivize survey participation with extra credit; unsurprisingly, response rates were much higher at those institutions. Furthermore, class sizes were not consistent. The baccalaureate courses ranged from 5 to 55 students apiece, while the doctoral university's courses averaged close to 400 students. Lastly, some institutions offered the course over multiple terms or semesters. Over the 2019-2020 academic year, *Dive into Systems* was used in 23 separate course offerings.

Students from 22 course offerings responded to the survey. The final student dataset includes 969 responses, including 698 responses from the doctoral university, 259 responses from the baccalaureate colleges, and 12 responses from the community college. Due to the unequal distribution of students across different institution types and course types, we present aggregated student survey results in Figure 1.

Figure 1a shows the distribution of student responses to the question, "How helpful did you find *Dive into Systems* compared to other computer science textbooks that you have read?" Students were asked to rate the perceived helpfulness of the book on a Likert scale, with 1 corresponding to "much less helpful", 3 corresponding to "equally helpful" and 5 corresponding to "much more helpful".

Students overwhelmingly rated the textbook favorably, with 92.3% responding that that they found *Dive into Systems* equally or more helpful than other textbooks they have used and 49.6% rating the book as either more helpful or much more helpful than other texts. When asked to elaborate on their rankings, students highlighted the clarity of explanations and the numerous examples and figures. Said one student, *"The examples were very clear and concise. The material was not overwhelming and the pages were not over-saturated with information like most textbooks"*. Other students felt the writing style made the material engaging: *"This is the first textbook in computer science that has not been boring and where I have found all of the information to be important and thoroughly explained."*

Students who gave a more neutral response mentioned their relative inexperience with prior CS textbooks and that they lacked a basis of comparison. Some of the students who found the textbook less helpful mentioned their preference for interactive content, such as interactive textbooks (e.g. ZyBooks), YouTube videos, or getting the material from lecture or teaching assistants.

Figure 1b shows the distribution of responses to the question, "Has the electronic nature of the textbook helped you learn less or more than textbooks in other courses?" Students rate on a Likert scale the perceived impact of the electronic nature of the book on their learning, with 1 corresponding to "much less", 3 corresponding to "about the same" and 5 corresponding to "much more".

The vast majority of students (90.1%) indicated that the electronic nature of the book aided their learning either the same or more than textbooks in other courses, with 50.6% indicating that it helped either more or much more. Students who rated the electronic aspect of the book positively were enthusiastic about how easy the book was to search and how it was always available. One student explained, *"its [sic] easier to search things up, and [I] always have it with me since its on my computer."* Others felt the free nature of the book was extremely advantageous. *"Electronic textbooks help a lot and the fact that it was free took some financial burden off college"* said another. Students who reported that the electronic nature of the textbook helped them "less" or "much less" commonly stated a preference for physical textbooks.

Additionally, we surveyed students with open-ended questions asking what they liked about the textbook and what they would suggest as improvements.

We heard several common themes among the book's positive feedback. One was how readable and understandable the content is. Students said its content is *"clear and direct"*, that the *"text was broken up well into concise, digestible sections"*, that it was *"focused, clear, easy to understand, and more engaging than most textbooks"*, and it *"explains complex subjects in...a very simple concise way."* Several students also described how the plentiful diagrams and examples aided learning. *"Excellent use of engaging examples"*, said one student. Another described the figures, diagrams and examples as *"very helpful in terms of understanding conceptual materials"*. Finally, students

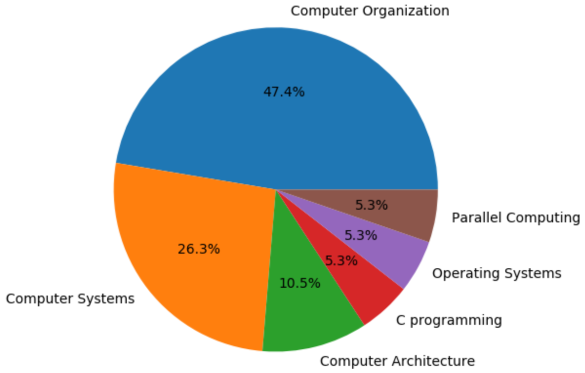


Figure 2: Types of Courses using *Dive into Systems*

commonly mentioned appreciating the book's free availability and the benefits of the book's format and accessibility.

By far, the most common improvement students suggested was to increase the number of exercises for measuring their understanding. One student mentioned that *"it would be nice to have a link to extra problems"*, another that wished for more *"problems at the end of chapters that can somehow check your answers."* Another related suggestion was for more interactive content with the online textbook. One student mentioned wanting *"...[m]ore interactivity. I'd enjoy quizzes or ways to practice what I was reading immediately."* Another suggested that *"some of the examples might be better displayed as a click-through animation."*

In another common theme, an emphatic minority of students lamented the lack of a physical textbook. These students also rated the electronic nature of the book as "less helpful" or "much less helpful". One student wrote, *"I will always be old fashioned and prefer a physical book."* Others mentioned how physical textbooks allow them to annotate content: *"I like to highlight important parts in the text to help me remember so I would print the pages for the reading"*. Others talked about their ability to "focus better" on physical textbooks, and the reduced eye strain of reading a physical text. *"I prefer a paper copy over an online copy because I don't like to look at a screen if I don't have to"* explained another student. To this end, we're working with *No Starch Press* to produce a print copy of the book for the subset of students who strongly prefer the traditional textbook format.

Finally, several students who used the book in fall 2019 wished for "key takeaway" chapter summaries, which we subsequently incorporated.

Overall, students highly praised the textbook. One student described it as a *"[g]reat textbook with a lot of helpful, well explained material"* another said *"the best textbook I have ever used honestly"*. Others said its language is *"friendly yet academic"* and that it *"isn't dry like many textbooks are"*, another saying that they *"fell in love with this book."*

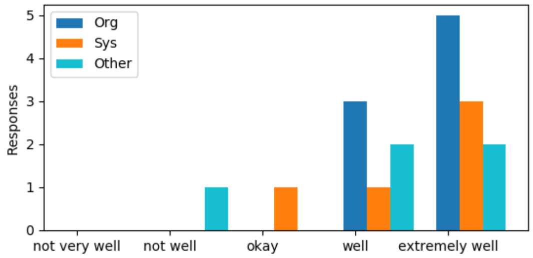


Figure 3: How well do *Dive into Systems* topics fit course?

5.2 Faculty Survey Results

We received a total of 23 faculty survey responses collected from 19 unique individuals; some faculty used the book in multiple terms and gave us feedback at the end of each term. We note that Figures 2 and 3 are over the 19 unique faculty respondents, while Figure 4 is over 23 per-course responses, as some faculty members' opinions changed between uses. We also note that none of the authors participated in the faculty survey.

Figure 2 shows the types of courses that used *Dive into Systems* in the 2019-2020 academic year. While the book was used for some specialized courses, the vast majority (73.7%) were either computer systems or computer organization courses.

We asked faculty to rate how well the topics in *Dive into Systems* fit their respective courses. Figure 3 shows the result. The vast majority of the faculty surveyed (89.4%) felt that the topics in *Dive into Systems* matched their courses either well or extremely well. Faculty who rated the fit positively commonly mentioned how the book *"focuses on similar learning outcomes"* of their courses, and the *"clarity of explanations"*. For example, one faculty member commented that *"the material was at exactly the correct level and corresponded exactly to the topics that I teach in my course"*. The faculty member who responded "not well" did not provide any detailed feedback; however, we speculate that since the faculty member in question ran an architecture course and used chapters of the book that were still under heavy revision in Fall 2019, it is possible that some of his concerns have since been allayed. The faculty member who gave an "okay" response mentioned that his institution used another introductory language (not Python) in their CS1 course; as a result, he needed to give extra context to students who were learning C for the first time. However, this faculty member also mentioned that he liked the book and planned to incorporate it more heavily in his course in upcoming semesters.

Next, we asked faculty to reflect on how *Dive into Systems* compares to other textbooks that they have used for their course. We note that since some faculty used the book in both the fall and spring terms/semesters, that the number of responses is higher in this question (23 total). The result is shown in Figure 4. The vast majority of responses (95.6%) believed that *Dive into Systems* was either equally as good or a better option for their course, with the majority (78.3%) feeling that *Dive into Systems* was either a better or much better option for their course. Lastly, we note that there is certainly a change in the distribution of responses between the Fall and Spring terms, with more faculty surveyed in the Spring seeing the book in a very positive light than those surveyed in the

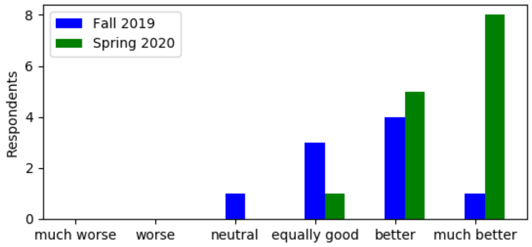


Figure 4: How does *Dive into Systems* compare to other texts?

Fall. We believe that a large part of this difference had to do with the continuous improvements to the book. In addition, four faculty used the book over both semesters. Two of the faculty gave a “much better" option for both semesters. The other two faculty improved their scores from "better" to "much better" and "equally good" to "better".

We also asked faculty to reflect on what they liked about *Dive into Systems*. Common themes in the responses were the quality of writing, fit to existing courses, use of examples and modern tools, and that it is free. *"The book explains the content well and focuses on similar core learning outcomes as what we try to each in the course,"* said one faculty member. Another faculty member mentioned that a free textbook enabled students to spend money on other materials that enhanced the learning experience in the course: *"I felt like Dive into Systems was almost designed for my course. Lots of great stuff that we needed, rather than having to buy multiple books. We spent the money on lab kits instead to do C programming with sensors on the Raspberry Pi."* Another faculty member simply remarked, *"Approachable, and readable content. Great figures."*

Since our early adopters were all at U.S. institutions, we asked faculty who taught in Spring 2020 to also comment on how *Dive into Systems* helped them (if at all) adjust to the remote teaching and learning environment during the COVID-19 pandemic. Several faculty explained that using an online textbook did not hinder the transition to a remote teaching environment, as the readings were already online. Others remarked that using *Dive into Systems* was vital to transitioning to the online environment. *"Thank heavens for the book"*, remarked one faculty member, *"It's the only thing that allowed me to function with the online course, since I could assign chapters/sections since they were so easy to read"*. These experiences are consistent with those at the authors' home institutions. For example, students who left campus during spring break typically did not travel with their textbooks. Consequently, many faculty were left scrambling to figure out how to get students the requisite reading materials for the remainder of the spring semester. Courses that had online readings had a much easier time with the transition, although some faculty mentioned that they couldn't explore the book as fully as they had hoped due to how hectic the Spring semester became.

Lastly, we asked faculty to reflect on what can be improved about *Dive into Systems*. The addition of a greater number of example problems and exercises in the text was the most frequent suggestion. Admittedly, the authors have focused primarily on writing the main text of the book for the last two years; exercises are a major focus point for future work. While the authors of *Dive into Systems* provided a preliminary instructor resources portal to early adopters, many of the surveyed faculty wanted to see more content in that portal, including labs, suggestions for projects, and additional exercises and solutions. A couple of faculty also mentioned adding interactivity to the textbook, so that students can run exercises interactively in the browser. This is another avenue of exploration for future work.

6 CONCLUSION AND FUTURE DIRECTIONS

We introduce *Dive into Systems*, a free, online textbook that presents a broad rage of computer systems, computer organization, and parallel computing topics for an audience with only a CS1 background. By releasing the book online at no cost, *Dive into Systems* supports the vision of making CS readily available to all students, particularly those who cannot afford traditional textbooks.

Our book is peer-reviewed and edited by volunteer experts from the CS education community. Through our 2019-2020 early adopter program, a beta version of the book was piloted by over 1200 students at over 18 institutions across the United States. Survey results show overwhelmingly positive feedback from both students and instructors. Students praised the clear and concise presentation of the book's content and appreciated the numerous examples and ease of searchability of the text. Instructors applauded how the book's topic coverage allowed them to easily match their course content and provided a much better alternative for their courses. Finally, both faculty and students appreciated the free online nature of our book, which we believe increases its likelihood of adoption.

We've made a *Dive into Systems* 1.0 release candidate freely available at *diveintosystems.org*. The online nature of the book allows us to easily incorporate improvements. As future work, we plan to produce interactive examples and more exercises, as many students and faculty asked for such materials in their survey feedback. We also plan to expand the faculty portal with additional materials. We're also working with *No Starch Press* to publish a print edition in conjunction with the final 1.0 release.

ACKNOWLEDGMENTS

We thank faculty who participated in our early adopters program and volunteers who helped edit the beta version of our textbook chapters (listed at *diveintosystems.org*). This work is partially funded through a 2019 SIGCSE Projects grant. The views expressed in this article are those of the authors and do not reflect the official policy or position of the Department of the Army, Department of Defense or the U.S. Government.

REFERENCES

[1] ACM/IEEE-CS Joint Task Force. 2013. Computer Science Curricula 2013. www.acm.org/education/CS2013-final-report.pdf.

[2] Alex Edgcomb and Frank Vahid and Roman Lysecky and Susan Lysecky. 2017. Getting Students to Earnestly do Readying, Studying, and Homework in an Introductory Programming Class. In *ACM Technical Symposium on Computer Science Education (SIGCSE)*.

[3] Allen Downey. 1999. *Think C++*. Green Tea Press. https://greenteapress.com/wp/think-c/

[4] Allen Downey. 2015. *Think OS*. Green Tea Press. https://greenteapress.com/thinkos/

[5] Andy Lynch and Brooke Ratto. 2012. Undergraduate Student Perceptions of a Free Textbook Alternative. *Business Education & Accreditation* 4, 1 (2012).

[6] Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau. 2020. Operating Systems: Three Easy Pieces. pages.cs.wisc.edu/~remzi/OSTEP.

[7] Avi Silberschatz and Peter Baer Galvin and Greg Gagne. 2012. *Operating System Concepts (9th Edition)*. Wiley.

[8] Barbara Ericson and Bradley Miller. 2020. Runestone: A Platform for Free, Online, and Interactive Ebooks. In *ACM Technical Symposium on Computer Science Education (SIGCSE)*.

[9] Brian L. Lindshield and Koushik Adhikari. 2013. Online and Campus College Students Like Using an Open Educational Resource Instead of a Traditional Textbook. *MERLOT Journal of Online Learning and Teaching* 9, 1 (March 2013).

[10] Codio. 2020. https://www.codio.com/.

[11] David Patterson and John Hennessy. 2013. *Computer Organization and Design (5th ARM Edition)*. Morgan Kaufmann.

[12] David Tarnoff. 2007. *Computer Organization and Design Fundamentals*. lulu.com.

[13] Emilie O. Falc. 2013. An Assessment of College Students' Attitudes towards Using an Online E-textbook. *Interdisciplinary Journal of E-Learning and Learning Objects* 9 (2013).

[14] Ian Wienand. 2020. Computer Science from the Bottom Up. https://www.bottomupcs.com/.

[15] John Hilton III and Carol Laman. 2012. One College's Use of an Open Psychology Textbook. *Open Learning: The Journal of Open, Distance, and e-Learning* 27 (2012). Issue 3.

[16] Lisa Petrides and Cynthia Jimes and Clare Middleton-Detzner and Julie Walling and Shenandoah Weiss. 2011. Open Textbook Adoption and Use: Implications for Teachers and Learners. *Open Learning: The Journal of Open, Distance and e-Learning* 26, 1 (2011), 39–49.

[17] Suzanne J. Matthews, Tia Newhall, and Kevin C. Webb. 2020. Dive into Systems. https://diveintosystems.org/.

[18] Noam Nisan and Shimon Schocken. 2008. *The Elements of Computing Systems: Building a Modern Computer from First Principles*. MIT Press.

[19] Randal Bryant and David O'Hallaron. 2015. *Computer Systems: A Programmer's Perspective (3rd Edition)*. Pearson.

[20] The NSF/IEEE-TCPP Curriculum Working Group. 2012. NSF/IEEE-TCPP Curriculum Initiative on Parallel and Distributed Computing - Core Topics for Undergraduates. http://www.cs.gsu.edu/~tcpp/curriculum/.

[21] Tu Do. 2020. Operating Systems: From 0 to 1. https://tuhdo.github.io/os01/.

[22] Umakishore Ramachandran and William Leahy. 2010. *Computer Systems: An Integrated Approach to Architecture and Operating Systems*. Pearson.

[23] Will Jarvis. 2019. A Textbook Giant Is Going 'Digital First.' That Might Not Be Good for Affordability. *The Chronicle of Higher Education* (July 2019).

[24] Yale Patt and Sanjay Patel. 2019. *Introduction to Computing Systems (3rd Edition)*. McGraw-Hill.