Teaching Statement

Adam Belay

I find teaching to be a highly rewarding experience and an essential component of my broader goals. I am interested in reducing developer effort and making computer systems more efficient. To this end, much of my research has focused on building the infrastructure needed to expose hardware features directly to applications. An equally important task is to train the next generation of students about the possibilities that are enabled by hardware and the techniques that are needed to use hardware effectively. In particular, I am excited to teach courses in computer architecture, operating systems, and networking.

My approach to teaching is closely tied to my research. For example, I like to draw on recent results to convey my sense of passion and excitement for computer science. Also, I tend to refer directly to original papers and source materials in my lessons to provide an intuition for how important ideas were discovered and how their usage has evolved over time. In a similar fashion, I appreciate how teaching can provide a valuable perspective that improves my research. For instance, I have found that the process of presenting course material and mentoring students helps to refine my understanding and bring important details into focus.

Teaching experience: While pursuing my Ph.D. at Stanford, I had the privilege of assisting with one course and co-instructing another. As a research assistant, I taught CS316, a graduate-level course on computer architecture. The course encouraged open-ended student research, exposing many students to the process of research for the first time. In addition to teaching recitation sections, I was responsible for mentoring student projects. My approach was to encourage students to discuss the topics that excited them most. I then connected these discussions back to related work and emphasized the importance of taking large enough risks to allow for potential improvements over the state of the art. To further encourage this type of thinking, I gave a guest lecture on systems software. It was a fantastic opportunity to relate computer architecture back to my research, and it inspired several students to undertake projects that spanned across hardware and software.

I took on even greater teaching responsibilities as a co-instructor (along with Prof. David Mazières and Ali Mashitizadeh) of CS140, an undergraduate introduction to operating systems. The course includes several labs and is designed to give students first-hand experience with systems software development. CS140 is very popular at Stanford; there were 142 registered students in my class. As a co-instructor, I was responsible for presenting a third of the weekly lectures. As part of my preparation, I worked to modernize the course material. For example, I updated the lecture on synchronization to include information about C11 atomics and recent results in kernel scalability. In addition to my teaching, I helped to manage a team of six teaching assistants. Administrating a course of this size was a significant undertaking, as even simple tasks like grading projects and exams required considerable coordination. I welcomed this challenge, and the course was a valuable opportunity for me to grow as a teacher.

Teaching plans: I am qualified to teach courses in operating systems, computer architecture, and networking, as well as introductory courses in computer science. My research often focuses on issues related to software scalability, and I would be very excited to create a special topic course on high performance I/O processing. It would be an excellent opportunity to give students first-hand experience in network protocols, scalable algorithms, and hardware device programming.

I am also interested in using my research as a tool to expose undergraduate students to basic operating systems concepts earlier in their computer science education. Operating systems is often a challenging course for students. Much of this difficulty stems from the need to peel back software layers so that it is possible manipulate hardware directly. My research project Dune provides a compelling alternative because it could be used to give students raw access to CPU features inside a familiar Linux environment. This approach would allow students to experiment directly with concepts like virtual memory without having to contend with the complexity or programming difficulty of a more traditional teaching operating system environment.
Mentoring: During the course of my Ph.D., I helped to advise several masters students and junior Ph.D. students. I believe that involving students in research is one of the most effective ways for them to develop engineering skills and gain experience with advanced concepts. One group of students that I mentored had an interest in my research project IX. Based on my guidance, they added support for network controller virtualization. The project was very successful, and has been fully integrated back into the broader system. As an advisor, I like to encourage students to identify and explore their own research interests. I am constantly amazed by the creativity and ingenuity of students, and it is has been a huge privilege to use my knowledge and expertise to help students gain experience in research.