I am excited about computer systems as an area because we leverage a set of core techniques in new and interesting ways to solve a wide range of problems. Applications and computing platforms are growing increasingly diverse and complex, with applications spanning data centers, mobile devices, and increasing numbers of IoT devices. Thus, students need to be able to solve complex computer systems problems. My goal as an advisor and teacher is to enable students — at every level — to identify problems and to address them with systems techniques.

I am interested in teaching classes on operating systems and computer networks at the undergraduate and graduate level; I can also cover introductory classes on machine organization and the hardware-software interface. My research background is broad enough to enable me to teach a graduate systems class for students in other areas. I can also co-teach special topics classes in the intersection of systems with other areas, for example, systems and programming languages. If the need arises, I can also teach other introductory undergraduate classes.

Advising For the past year, I advised undergraduate student Paul Yau on a research project; he is currently applying to graduate school. The project consisted of building an emulator for the network interface hardware model I was developing as part of my research. I found I needed to adjust my advising over the course of the project. Initially, I met with Paul for hour-long sessions, often twice a week, to discuss background material he had been working through. As Paul started working on implementation, our meetings focused primarily on implementation details until, with the project nearing completion, I switched to short weekly status meetings and providing high-level guidance as needed. Less frequent meetings with the faculty advisor on the project provided Paul with an opportunity to take a step back and think about the bigger picture when discussing status and next steps.

Based on this experience, I think it works well to pair each junior student with a senior student who can serve as a mentor and collaborator. For students without peer mentors, I can provide hands-on advising. A co-advisor can help students benefit from different advising styles. To help graduate students grow into independent researchers, individual projects and collaboration are both important. Individual projects encourage students to take responsibility for their work while collaboration on other projects exposes students to new ideas and distributes the workload during critical project phases.

Teaching Over the course of my studies, I served as a teaching assistant for different types of computer systems classes at every level. In my view, project experience is invaluable, students in the project-based class I helped teach came away from the class with a deeper understanding of the concepts, as well as the benefit of hands-on experience in implementing and debugging computer systems components. This also matches up with my observations as a corporate training instructor for computer networks as well as my personal experience as a student learning those concepts.

Thus, I plan to build my courses around class projects as much as possible, to provide students with a deep understanding as well as practical experience with systems concepts. For undergraduate courses, course projects should have a set of milestones where each milestone requires implementing a clearly defined component of the system. For example, an undergraduate operating systems class could consist of implementing parts of OS subsystems including threads, synchronization primitives, and virtual memory management. By splitting class projects into multiple milestones I can track student progress over the term and address problems before the end of the term. At the graduate level, clearly defined milestones help students to get started but need to be combined with a larger open-ended project allowing students to explore a systems problem of their choice.