Teaching Statement

Marios Kogias

One the main motives for me pursuing an academic career is the chance to work with students and help them evolve. My philosophy behind teaching and mentoring is that each student has a different set of skills and talents and its the educator's goal to help them explore those and use them either individually or as part of a team.

During my time as a PhD student at EPFL, I had the opportunity to mentor a lot of students and work as a teaching assistant under different professors which helped me identify and use different elements of the teaching methodology in order to start forming my own identity as an educator. My approach to teaching and mentoring depends on the following three pillars. Enhancing the students' **reading and writing** abilities in the context of scientific literature is crucial for students to comprehend fundamental principles but also to express themselves and distil their own ideas. I strongly believe that writing is a fundamental part of the design process. **Experience-based learning** is a key way to engage and motivate students to start exploring and learning alone. Experience can come through projects or just real-world examples that relate to the topic under discussion. Finally, **group-based learning**, either through group projects or group discussions during the lecture, apart from boosting collaboration, enables students bring their own skills to the group and puts them in a teaching position within this group thus fostering those skills even further but also motivating the rest of the group to participate and evolve.

Mentoring Experience

As part of the Datacenter Systems Laboratory at EPFL I supervised 5 master thesis, 7 master projects, and 2 bachelor projects in the 5-year course of my PhD. You can find all the student projects and their reports in my website[1]. I tried to group the offered projects in three main categories trying to accommodate different groups of students. The first category, targeting mostly students that were interested in industrial positions, included projects that were pretty well defined and they required certain skills. The goal of these projects was for the students to contribute to an existing code base and have their code eventually merged. Example of such a student is Mikael Morales where he contributed to the Lancet load generator. The second category, targeting mostly master students interested in doing research or going for a PhD, included more research-y and open-ended projects that required the student to come up with a new design for the given problem, do the equivalent literature search, and create a proof of concept. Example of such a project is Konstantinos Prasopoulos' thesis on congestion control for R2P2, who after that started his PhD at EPFL. Finally, the last category targeted students that were interested in learning a new technology without having prior knowledge in it. The students were required to do the relevant search to understand this technology and use it in a small project. Example of such a project is Mattheu de Beule's work on eBPF.

The general approach I had with my students was the following. Initially, we had a meeting to discuss more about the project, their interests, how the project aligns with their skills, potentially alter the project accordingly, and set the project goals. After that, I liked having weekly meetings with the student. These meetings had different forms based on the student needs Certain students wanted to have a clear task set that we agreed on every meeting, while others liked to be more independent so we had a more general discussion about the problem they are tackling. Apart from the organised meetings many times we had special purpose meetings where we worked together on the codebase and discussed about the design or the performance analysis of their project or generally about research directions based on their topic of interest. All of my students had to submit a report at the end of the project to also improve their writing skills and learn to incorporate writing as part of their problem solving, and do a presentation during the lab weekly meeting. All my students were part of the DCSL during their project so that they feel included and benefit from the lab culture. This environment

[1] https://marioskogias.github.io/teaching/

encouraged students to come back for another project or work on a different topic that they explored during their time in the lab. Such an example is Martin Weber who worked on Clang and scheduling for his master project, but switched to programmable switches and P4 for his master thesis.

Teaching Experience

As a PhD student at EPFL I served as a teaching assistant for different courses, both bachelor and master level. At a bachelor level I TA-ed introductory courses to computer networks (COM-208) and operating systems (CS-323) where I was in charge of weekly exercises and labs. Working with undergrads helped me improve my ability to explain fundamental principles to students without prior background and make the lecture engaging by bridging the course material with real use cases. For example, during an office hour session for COM-208 we built a simple NAT middlebox using iptables so that students could fully comprehend NAT and where it is used instead of just learning the theory behind NAT. At a master level, I TA-ed Principles of Computer Systems (CS-522) several times and Database Systems (CS-422). As a POCS TA I was in charge of leading a weekly paper discussion session and create one pager assignments in which students had to come up with a design to apply the principle under discussion in an open-ended problem. To enhance students writing and comprehension skills and to underline the importance of abstraction I introduced a new kind of assignment for POCS in which students had to write a paper summary in a page, in a paragraph, and in a sentence. This significantly helped them understand the papers under discussion and reason about the underlying ideas.

In my last year at EPFL I organised the systems reading group. The goal of this group was to track on-going research in computer systems by reading and discussing recent papers after each conference. Every week there was a discussion leader and the group had one main rule: everyone had to ask at least a question or make a comment on the paper so that even junior students are encouraged to participate in the discussion. Having people with different backgrounds and different research interests discuss a paper whose topic could be someone's exact research focus while others are just familiar with the area was very beneficial for everyone both in order be up-to-date with recent research but also to get different perspectives for a research problem.

Despite irrelevant to computer science, but highly relevant to teaching, during my undergrad and graduate studies I worked as a dance instructor both for adults and elementary school children. Teaching how to dance to such different groups of people helped me understand how to adapt the same teaching material to the dynamics of the class and the personal interests students might have. Working on student engagement and making students a major part of the teaching process rather than passive consumers was a key point in my approach especially when dealing with elementary school children.

Teaching Plans

Based on my research experience and interests, I am interested in teaching classes related to computer networks, operating systems, and distributed systems at a graduate or undergraduate level. Also, I could teach introductory computer science courses, such as introduction to programming.

Given the chance I would like to introduce a course on advanced practical networking where students will have a deep dive in network stacks and their implementation across different technologies, such as kernel-based networking, kernel-bypass, eBPF, and P4. The class will revolve around a main project will require students to design a new toy network protocol and implement it end to end, from the user API, to the actual host side implementation, and the involved software and hardware middleboxes. Throughout the course apart from the hands-on experience with different networking technologies students will learn basic system design principles and acquire the necessary performance evaluation skills to compare their implementation with the state of the art and the implementation of their peers that should be interoperable.

Another course that I would like to introduce is a course on datacenters. Datacenter systems are highly customizable and divert a lot from general purpose computing. The goal of the course should be for students to understand the basic evaluation metrics that are used and specifically target datacenters, such as tail latency, tail tolerance, and energy efficiency, and how optimising for those metrics affects system design inside the datacenter. The format of the course will be similar to a reading group and includes a weekly discussion of seminal datacenter systems while the major part of the grade will come from a group project on a topic relevant to the course material.