Marco Perronet

PhD Student

24th November 1997

Italian

github.com/perronet

linkedin.com/in/marco-perronet

perronet@mpi-sws.org

About me

I am second year PhD student in computer science at Max Planck Institute. I am interested in operating systems, networking, and distributed systems, but I am also open to fields that I am less comfortable with.

I moved to Germany after my Bachelor to pursue a **Master+PhD** program at Max Planck Institute: in this program, students study for courses while working on research, and then continue the PhD after graduation. I have recently completed my master courses and I am currently focusing on my research, aiming to obtain my first scientific publication.

Languages

- **Italian** (mother tongue)
- **French** (secondary language, I am bilingual by birth)
- **English** (very fluent)
- **German** (intermediate)

Skills

I have programming experience in **Java**, **Rust**, **Python**, and **Haskell**. Being interested in operating systems and a long-time **Linux** user, I have also extensive experience with **Bash** scripting and **C** programming. I am also confident with versioning control systems based on **Git**.

During my two years of **research** I learned how to tackle complex problems. I can design, document, and present my solution, then implement and evaluate it. This experience also taught me how to deal with open-ended projects: from defining goals to measuring progress and pivoting when necessary.

Education

since 2019 — **PhD student** — Max Planck Institute For Software Systems
Real-time systems team

since 2019 — **Master's degree** — Technische Universität Kaiserslautern
Expected in 2021

2016-2019 — **Bachelor's degree** — Università degli studi di Torino
Thesis title:
"Linux Kernel: monitoring the scheduler by `trace_sched*` events"

Research

I am currently working in the field of Real-time Systems. My research is on operating systems and my work is focused on trace-based response-time analysis on Linux.

2020 — **RBF Trace**

A tool in Rust that I am developing as part of my ongoing project. Being my main project, I created the design and wrote the majority of the code while coordinating a small team of four people.

RBF Trace can trace the processes running on the system, extract a formal model, and use it to analyze the timing correctness of the system, which is a crucial aspect of real-time systems. Unlike existing tools, RBF Trace performs the analysis at runtime on the analyzed system, and the behavior of processes is inferred from the trace.

Projects

2019 — **Linux kernel exploration**

In the recent years, I have been fiddling with the kernel codebase both for my reasearch work and personal interest. Last year I contributed to the kernel by showing the existence of a minor bug in the real-time scheduler. I then collaborated with a kernel developer to fix it, and the patch is now on the way for merge (more information here).

2018 — **NP to SAT transform**

In this project, I implemented in C the Cook-Levin theorem from complexity theory. This tool takes as input an arbitrary problem in the form of a Turing machine and transforms it into an instance of the SAT problem (boolean satisfiability problem). A SAT solver then tries to solve the formula, and it will manage to do so only if the Turing machine would give a positive answer for the original problem.

2016 — **Interactive 3D map**

I 3D mapped the building of my high school and implemented it as a playable map as part of a videogame built with Valve's Source Engine. In this project, I learned about level design and 3D modeling with Blender.