Chair of Decentralized Systems Engineering
Department of Informatics

Cross-ISA Binary Format for Efficient Binary Translation

General information	**Advisor** Dr. Redha Gouicem **Email** gouicem@in.tum.de **Date** 25.01.2022
Type	Master / Bachelor
Description	With the advent of new Instructions Set Architectures (ISA) such as ARM and RISC-V in server and consumer markets, application compatibility is threatened. Indeed, binaries compiled for x86 cannot be executed on these new architectures, and need to be recompiled. However, the source code is not always available, and build dependencies problems can arise on the new ISA. Binary translation is a solution to this problem, by allowing this cross-ISA execution by translating the binary from one instruction set to the other. Binary translation exists in two flavors: static and dynamic. Static translation takes a binary and translates every instruction to the new ISA, creating a new binary. Dynamic translation executes a binary while translating it bit by bit. The former allows for better generated code, but has code coverage issues. The latter, however, can translate everything, but produces low quality code. **Project:** The project we are currently working on aims at combining these two flavors and developing a hybrid binary translator. The idea is to perform static translation first, thus generating a well optimized binary

	through LLVM [1]. However, this binary will have holes, since x86 cannot be fully translated without dynamic information only available at run time. This binary will then be used in a dynamic binary translator based on QEMU [2], that will execute the binary and fill the holes when necessary. The translated instructions will also be saved in the binary for future reuse and for offline optimization. **Thesis:** Your main task will be to design and implement an ELF-based binary format [3] that will be used across the static and dynamic translators. Indeed, the binary should be able to hold both source and target ISA instructions [4]. You will need to modify the static binary translator to produce a binary in that format, as well as enable it to read the format for future optimization of the code added by the dynamic translator. You will also need to modify the dynamic translator to enable execution of binaries in this new format as well as appending newly translated instructions.
Keywords	Static binary translation, dynamic binary translation, binary format, LLVM, QEMU, ELF
Goals	**Concrete outcomes** 1. Design and implementation of an ELF-based cross-ISA binary format 2. Implementation of this binary format in the static translator 3. Implementation of this format in the dynamic translator 4. Evaluation of the new binary format (binary size, impact on performance) **Bonus points** 5. Extend the binary format to include intermediate representation 6. Implement optimizations based on the intermediate representation
Prerequisites	**Compulsory** ● Proficient in C/C++ **Preferred** ● Knowledge of QEMU ● Knowledge of LLVM

References

1. QEMU
2. LLVM
3. ELF Binary format
4. FatELF: Universal Binaries for Linux

Application process

Please send an email to the advisor including the following:

- Email subject: "Thesis application (DSE)"
- CV
- A copy of your transcript(s)
- A **motivation statement**, please include samples of your work that you are proud of (e.g., major projects, open-source contributions, Github page, etc.) and/or writing samples (e.g., your technical blog, project reports, etc.)