Contents

List of Scripts

List of Figures

Acronyms

API Application Programming Interface.

DSL Domain Specific Language.

HAL Hardware Abstraction Layer.

IAL Interrupt Abstraction Layer.

IPC Inter-Process Communication.

MAL Memory Abstraction Layer.

MMU Memory Management Unit.

OS Operating System.

SOS Structural Operational Semantics.

TCB Trusted Computing Base.

1. Introduction

The following report describes the activities carried out during a 12-week, full-time internship at the Research Center in Computer Science, Signal and Automatic Control of Lille (CRIStAL). This internship revolves around the PIP pretokernel and DEC currently being developed by the 2XS team. PIP is a minimal OS kernel with provable memory isolation using Hoare logic. The deep embedding or DEC, as opposed to the shallow embedding, is an intermediate language for the translation of PIP to C. Specifying programs in the deep embedding has the great advantage of simplifying their syntactic manipulation. It also ensures a stricter structuring of program expressions. **Therefore, we want to compare deep and shallow proofs and, more precisely, check whether this structure is reflected in the proofs done in the deep embedding ?**

To that end, we will model three different functions in the deep embedding. The first one reads the memory. The second one writes in the memory. The last one is a recursive function. Then, we will prove invariants about these functions. One of the invariants propagates all of PIP's properties which include memory isolation, vertical sharing, kernel data isolation and consistency. We chose a modular approach in our implementation and we engineered ours proofs correspondingly. We also did some preliminary work mostly to get familiar with Hoare logic and the deep embedding.

The first part of the report offers an overview of the CRIStAL laboratory, the 2XS team and their research activities. The second part is dedicated to the PIP protokenerel and focuses on its proof oriented design, its properties, its data structures, Hoare logic theory and, more importantly, the deep embedding and its constructs. The last part is dedicated to our contributions, detailing how we modelled the required functions, how we engineered our proofs in the deep embedding and our observations about the comparison between the deep and shallow proofs.