Abstract

Software nowadays is often designed with dependability in mind. Our thesis combines two approaches in creating a more dependable system: using microkernels instead of monolithic kernels and formally verifying software. We have tried to verify three properties of inter-process communication in the Fiasco microkernel. As Fiasco has been written in C++, which does not support verification, we converted the source code to a model in the PVS proof system. To keep the model and proofs compact, we abstracted away many details of inter-process communication.

Two of the three properties were verified; both dealt with threads possibly waiting forever. The third property, verification of the assertions in the source code, posed several problems. One problem proved insurmountable, probably due to the abstractions applied. Another problem led to the finding of a bug in Fiasco's IPC implementation. Although finding the bug had clear, practical use, we consider the fact that our abstract model could find the bug more important. It shows that one does not have to create a one-on-one model to apply (partial) verification; even our model in which essential components were abstracted away sufficed to find a bug.