Abstract

Within the education sector, unit testing is commonly used to assess the correctness of programs, often to automatically grade students assignments and also offer rapid feedback to students. Within the context of teaching ARM Assembly, a low-level language, this is no different, though the tooling to test Assembly programs can often be cumbersome when writing complex tests.

This project addresses this issue by allowing tests to be written for ARM Assembly programs in a higher level language, in this case, Python. This involves designing and implementing a test harness which is suitable for use by novice programmers in an educational setting. In order to implement the test harness, an intermediary layer was built that provides an abstraction around an emulator so that the ARM Assembly programs could be executed on common hardware.

Being built for students and novice programmers, the test harness exposes intuitive APIs for invoking subroutines and supports interacting with registers and memory. As the goal of unit testing is to uncover and fix bugs in the code being tested, care was taken to ensure that any error messages that arise during execution are clear, readable and actionable.

Finally, a code coverage tool has also been developed to help assess the quality of a test suite.