

MINFUN LANGUAGE CODE GENERATOR

Part 3: Assembly Code Generator



April 17, 2013

Table of Contents

Introduction..... 3

Using the Mfun Interpreter..... 3

Challenges 3

Testing..... 3

Introduction

For this project, we are to write a code generator for the Minfun language. Minfun is a simplified implementation of the Scheme Language. In this design phase, we were responsible to generate assembly code from Minfun source files. The generator is run by passing a file as an argument to the mfunc shell script. The generator was validated by using JUnit.

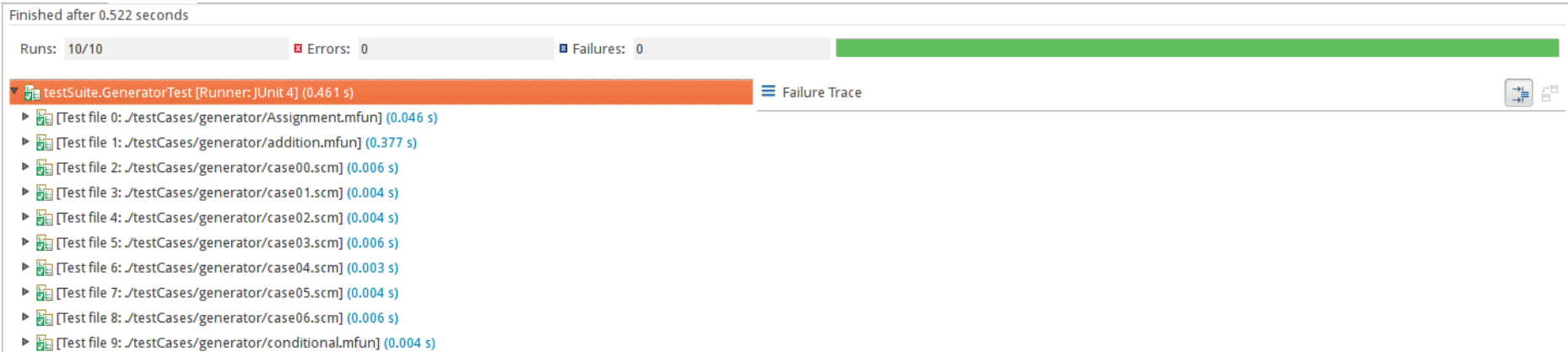
Using the Mfun Interpreter

This interpreter will only work in a Unix environment with Java, JavaCC and JTree installed. Run './mfunc <filename>'. Enter '2' to run the generator. The assembly output file will be written to the 'output' directory.

Challenges

- Learning how to push and pop from the stack using the MIPS architecture
- Coming up with test cases

Testing



I used JUnit to test my code generator by creating a test suite. I used 7 test cases which I created. I included test cases provided to us from Blackboard. One by one, each file path is passed as an argument to the Parser.parse() static method and then the result to the Generator.gen() method. The test case fails if the method throws either: a ParseException, or an IOException. The resulting assembly code was checked by using 'spim'.