Assignment 10: BF to LLVM

Cayden Lund (u1182408)

7 November 2024

Repository: https://github.com/caydenlund/brainforge

Benchmarks

For each of the following benchmarks, I compiled a version with and without simple loop optimizations using my regular BF compiler, `bfc`. I also compiled a version with and without simple loop optimizations using my BF-to-LLVM compiler, under optimization level `-O0`, `-O1`, `-O2`, and `-O3`. I ran each compiled binary 100 times and recorded the execution time.

These tests were all done on the same machine, a laptop with a Ryzen 5 5500U CPU and 16 GB of RAM. The median runtime is shown below.

Program	Median `bfc` Runtime		Median `bf-llvm -O0` Runtime		Median `bf-llvm -O1` Runtime		Median `bf-llvm -O2` Runtime		Median `bf-llvm -O3` Runtime	
	Base	Simple Loops	Base	Simple Loops	Base	Simple Loops	Base	Simple Loops	Base	Simple Loops
`bench.b`	0.235s	0.001s	0.308s	0.002s	0.237s	0.001s	0.237s	0.001s	0.237s	0.001s
`bottles.b`	0.000s	0.000s	0.001s	0.000s	0.001s	0.000s	0.001s	0.000s	0.001s	0.000s
`deadcodetest.b`	0.000s	0.000s	0.000s	0.000s	0.000s	0.000s	0.000s	0.000s	0.000s	0.000s
`hanoi.b`	3.784s	0.042s	4.897s	0.093s	3.872s	0.052s	3.873s	0.052s	3.875s	0.052s
`hello.b`	0.000s	0.000s	0.000s	0.000s	0.000s	0.000s	0.000s	0.000s	0.000s	0.000s
`long.b`	3.160s	0.200s	4.598s	0.624s	3.457s	0.399s	3.469s	0.399s	3.458s	0.399s
`loopremove.b`	0.000s	0.000s	0.000s	0.000s	0.000s	0.000s	0.000s	0.000s	0.000s	0.000s
`mandel.b`	0.792s	0.747s	1.104s	1.036s	0.917s	0.878s	0.918s	0.889s	0.929s	0.870s
`serptri.b`	0.000s	0.000s	0.000s	0.000s	0.000s	0.000s	0.000s	0.000s	0.000s	0.000s
`twinkle.b`	0.000s	0.000s	0.000s	0.000s	0.000s	0.000s	0.000s	0.000s	0.000s	0.000s

Interestingly, my compiler outperformed the LLVM binaries, even at high optimization levels. I intend to do some experimentation with allowing extra passes to see how that changes things.