Samuel Calise and Claudia Deverdits | CSC411 Profiling Assignment Lab Notes

Benchmarks	Time (s)	# of Instructions	Rel to start	Rel to prev	Code Improvements
midmark.um	Real 1.586s User 1.554s Sys 0.001s	10048	Real 1.000s User 1.000s Sys 1.000s	Real 1.000s User 1.000s Sys 1.000s	Initial test of program, no improvements
sandmark.um	Real 39.614s User 39.080s Sys 0.088s	10046	Real 1.000s User 1.000s Sys 1.000s	Real 1.000s User 1.000s Sys 1.000s	
midmark.um	Real 0.813s User 0.781s sys 0.016s	10048	Real 0.512s User 0.502s Sys 16.00s	Real 0.512s User 0.502s Sys 16.00s	Instead of using the bitpack, we have improved our code to have the getu function into the um_instruction.rs file. And added unsafe and get_unchecked_mut into get_register_value function in register.rs
sandmark.um	Real 20.926s User 20.617s Sys 0.024s	10046	Real 0.528s User 0.527s Sys 0.272s	Real 0.528s User 0.527s Sys 0.272s	
midmark.um	Real 0.588s User 0.568s Sys 0.017s	10048	Real 0.370s User 0.365s Sys 17.00s	Real 0.723s User 0.727s Sys 1.062s	Change the main.rs “for” loop to “loop” in addition to making the function calls to rum in a match statement when matching the correct instruction opcode value to the respected function call.
sandmark.um	Real 16.144s user 15.885s sys 0.065s	10046	Real 0.407s User 0.406s Sys 65.00s	Real 0.771s User 0.770s Sys 2.708s	
midmark.um	real 0.614s user 0.594s sys 0.004s	10048	Real 0.387s User 0.382s Sys 4.000s	Real 1.044s User 1.045s Sys 0.235s	In rum.rs made the get_instruction function unsafe based on this function being called multiple times according to the flamegraph
sandmark.um	real 15.149s User 14.953s Sys 0.033s	10046	Real 0.382s User 0.382s Sys 0.375s	Real 0.938s User 0.941s Sys 0.507s	
midmark.um	real 0.625s user 0.619s sys 0.001s	10048	Real 0.394s User 0.398s Sys 1.000s	Real 1.017s User 1.042s Sys 0.250s	In rum.rs, we created a tuple of A,B, and C from the instruction and still kept the later implementation in the function’s definition. This resulted in a slight improvement.
sandmark.um	real 15.113s user 14.961s sys 0.037s	10046	Real 0.381s User 0.382s Sys 0.420s	Real 0.997s User 1.001s Sys 1.121s	

midmark.um	real 0.537s user 0.532s sys 0.001s	10048	Real 0.338s User 0.342s Sys 0.001s	Real 0.859s User 0.859s Sys 1.000s	Slightly faster time by adding “#[inline]” to some functions within rum cargo
sandmark.um	real 14.001s user 13.799s sys 0.029s	10046	Real 0.353s User 0.353s Sys 0.329s	Real 0.926s User 0.922s Sys 0.783s	
midmark.um	real 0.500s user 0.493s sys 0.005s	10048	Real 0.315s User 0.317s Sys 5.000s	Real 0.931s User 0.926s Sys 5.000s	Improved map_segment so unsafe is used when mapping a segment to a new address
sandmark.um	real 13.952s user 13.758s sys 0.021s	10046	Real 0.352s User 0.352s Sys 0.238s	Real 0.996s User 0.997s Sys 0.724s	

After trying to incorporate all class files into main.rs and testing three compile attempts to potentially achieve a lower score, we were unable to show a lower runtime. The above lab notes represents the most optimal and greatly improved code structure based on your initial time records for our submission.