

Benchmark	Time(Seconds)	Instructions	Rel to start	Rel to prev	Improvement
Sandmark	256.63s	-	1.000	1.000	Initial state
Midmark	10.30s	59,023,600,230	1.000	1.000	
Sandmark	98.65s	-	0.38	0.38	Compiled with optimization 01 turned on
Midmark	3.96s	31,784,014,387	0.38	0.38	
Sandmark	101.05s	-	0.39	1.02	Compiled with optimization 02 turned on (discarded)
Midmark	4.04s	29,449,380,954	0.39	1.02	
Sandmark	74.44s	-	0.29	0.75	Fine-tuned Bitpack_getu and bit shift helpers
Midmark	2.98s	17,642,290,107	0.28	0.73	
Sandmark	71.58s	-	0.27	0.96	Tuned opcode and registers reading
Midmark	2.89s	17,642,290,107	0.28	0.97	
Sandmark	69.3s	-	0.27	0.96	Replaced Bitpack_getu calls with implicit bit shifts
Midmark	2.69s	11,496,075,429	0.26	0.93	
Sandmark	48.91s	-	0.19	0.70	Revamped memory segment representation*
Midmark	1.94s	8,372,840,981	0.18	0.72	
Sandmark	18.79s	-	0.07	0.38	Revamped memory representation**
Midmark	0.74s	4,012,800,103	0.07	0.38	
Sandmark	14.56s	-	0.05	0.77	Specialized function calls and tuned code in main interpretation loop.
Midmark	0.57s	3,449,860,149	0.05	0.77	
Sandmark	12.96s	-	0.05	0.89	Revamped unmapped memory segment representation***
Midmark	0.50s	3,170,451,263	0.04	0.87	

All recorded times are averages of 3 runs.

Final benchmark on advent.umz with partial solution: 6.181s

*I had originally utilized Hanson's Array for my inner memory segment representation with the intent to utilize the provided copy function, as that is the only functionality these inner segments really need. I replaced this with a struct with utilizes a uint32_t pointer and an unsigned integer. The pointer represents the segment and is now calloc'ed, and the unsigned integer is used to keep track of the segment size as that information is needed to preform a copy. I implemented a resize function for this struct as well.

**I had originally used Hanson's Sequence for my outer memory segment representation with the intent to utilize the built in resize-ability of this data structure. I switched this out for a struct of struct pointers(of the kind I defined above, the 'inner segments') which represent the sequence of instructions and an unsigned integer to keep track of the size. I built a resize function for this struct as well.

***I had originally used Hanson's stack for my unmapped segments. I switched this out for a struct of uint32_t pointers which represent the unmapped segment IDs and both an index to keep track of the current number of unmapped IDs and a size. I also implemented a resize function for this struct.