Easton Kang (ekang05), Jonathan Liu(jliu36)
COMP-40
Professor Monroe & Townsend
12/1/2019

Profiling Record

Midmark

Stage	Execution Time	Instructions	Time Ratio To Prev	Time Ratio to Start	Improvements
No Changes	2.50s	26,172,636,535	1	1	No improvements (starting point)
Link with -O1	2.36s	19,162,481,615	0.944	0.944	Linked with -O1
Link with -O2	2.35s	25,529,148,143	0.995	0.940	O2 is *usually* better than O1. O2 performs more elaborate optimizations
Change from Sequence to int	2.34s	25,527,100,382	0.996	0.936	Changed the mapped ID's sequence to a flat integer since the sequence was unneeded
Removed recursion in input()	2.32s	25,527,100,414	0.991	0.928	Removed input() recursive call. Instead calls fgetc explicitly instead of calling a new instance of input() to the stack
Program counter is no longer	2.28s	25,134,251,349	0.983	0.912	There was no need to malloc the program

allocated in the heap					counter. Rather it is allocated in the stack
Compiled with -O3	1.47s	18,425,184,867	0.645	0.588	Compilation with flag O3 maximizes optimizations
Remove intermediary functions	1.45s	18,383,126,827	0.986	0.580	Removes segmem.c to reduce the amount of function calls from different files
Remove intermediary functions for operations	1.44s	18,327,574,438	0.993	0.576	Moves basic operations NAND, cmov, add, divide, mul, load value into the main file
Remove intermediary functions and unnecessary structs	1.39s	17,219,899,528	0.965	0.556	Moves main instruction code into the main while loop, and removed a struct that held registers
Move the rest of instructions into main	1.31s	14,292,378,311	0.942	0.524	Moves the rest of the functions in instructions all into main.
Used a different bitpack function	0.45s	4,430,784,937	0.344	0.180	The given bitpack was using 64% of performance. We imported the Bitpack_get

					function

Sandmark

Stage	Execution Time	Instructions	Time Ratio To Prev	Time Ratio to Start	Improvements
No Changes	64.52s	649,157,777,864	1	1	No improvements (starting point)
Link with -O1	61.02s	663,186,280,269	0.946	0.946	Linked with -O1
Link with -O2	61.73s	633,186,280,555	1.012	0.957	O2 is *usually* better than O1. O2 performs more elaborate optimizations
Change from Sequence to int	61.06s	633,172,778,233	0.989	0.946	Changed the mapped ID's sequence to a flat integer since the sequence was unneeded
Removed recursion in input()	61.01s	634,225,549,218	0.999	0.946	Removed input() recursive call. Instead calls fgetc explicitly instead of calling a new instance of input() to the stack
Program counter is no longer allocated in the heap	59.82s	623,414,617,543	0.980	0.927	There was no need to malloc the program counter. Rather it is allocated in the stack
Compiled with	37.58s	456,745,784,8	0.645	0.583	Compilation

-O3		66			with flag O3 maximizes optimizations
Remove intermediary functions	36.24s	443,256,235,353	0.964	0.562	Removes segmem.c to reduce the amount of function calls from different files
Remove intermediary functions for operations	36.14s	429,127,462,325	0.993	0.576	Moves basic operations NAND, cmov, add, divide, mul, load value into the main file
Remove intermediary functions and unnecessary structs	35.08	426,876,413,402	0.965	0.556	Moves main instruction code into the main while loop, and removed a struct that held registers
Move the rest of instructions into main	32.78s	400,467,332,849	0.934	0.508	Moves the rest of the functions in instructions all into main.
Used a different bitpack function	11.23s	109,065,642,437	0.343	0.174	The given bitpack was using 64% of performance. We imported the Bitpack_get function