

Comp40 hw7 - profiling  
Duke Alf and Evan Loconto  
(ealf01 and elocon01)

Benchmark	Time (s)	instructions	Rel to start	Rel to prev	Improvement
Midmark adventure Sandmark	6.747 55.490 169.754	$3.033 \times 10^{10}$ -- --	1.000 1.000 1.000	1.000 1.000 1.000	No improvement (starting point)
Midmark adventure Sandmark	5.190 44.401 130.859	$2.692 \times 10^{10}$ -- --	.769 .800 .771	.769 .800 .771	Compiled with optimization turned on and linked against -lcii -O1
Midmark adventure Sandmark	5.083 46.404 129.628	$2.689 \times 10^{10}$ -- --	.753 .836 .764	.979 1.045 .991	Compiled with optimization turned on and linked against -lcii -O2
Midmark adventure Sandmark	5.074 43.991 130.514	$2.689 \times 10^{10}$ -- --	.752 .793 .769	.998 .948 1.007	Removed seq_length call from for loop in free_memory and placed result in local variable instead. Little to no improvement, considering the for loop is only called once. Along with this, there is some timing discrepancy when using the server/error range of timing results.
Midmark adventure Sandmark	5.056 44.146 129.952	$2.689 \times 10^{10}$ -- --	.749 .796 .766	.996 1.004 .996	Removed seq_length call from for loop in duplicate_segment and placed result in local variable instead.
Midmark adventure Sandmark	4.932 43.498 125.514	$1.508 \times 10^{10}$ -- --	.731 .784 .739	.975 .985 .966	Remove use of Bitpack_get() in execute and use shifts instead
Midmark adventure Sandmark	3.148 28.504 79.080	$1.177 \times 10^{10}$ -- --	.467 .514 .466	.638 .655 .630	Replace segments as Sequences to segments as arrays
Midmark adventure Sandmark	2.083 17.999 51.314	$8.951 \times 10^9$ -- --	.309 .324 .302	.662 .631 .649	Replace memory as Sequence to memory as dynamic array
Midmark adventure Sandmark	2.022 17.281 51.177	$7.93 \times 10^9$ -- --	.300 .311 .301	.971 .960 .997	Get rid of decode() and put code directly into execute()

Midmark adventure Sandmark	1.891 15.987 52.037	7.93 x 10 <sup>9</sup> -- --	.280 .288 .307	.935 .925 1.017	Static Inline operation functions
Midmark adventure Sandmark	1.636 13.855 41.767	7.052 x 10 <sup>9</sup> -- --	.242 .250 .246	.865 .867 .803	Remove all operation functions and place code directly in switch statement
Midmark adventure Sandmark	1.507 13.275 38.504	7.052 x 10 <sup>9</sup> -- --	.223 .239 .227	.921 .958 .922	Get rid of initialize() and place code directly into execute()
Midmark adventure Sandmark	1.330 10.819 32.228	6.007 x 10 <sup>9</sup> -- --	.197 .195 .190	.883 .815 .837	Get rid of get_word() and place code directly where called in execute.c --get_word took up 17% of cachegrind
Midmark adventure Sandmark	1.241 10.915 31.905	6.003 x 10 <sup>9</sup> -- --	.184 .197 .188	.933 1.009 .990	Get rid of bitpack_newu() and now we never use bitpack
Midmark adventure Sandmark	0.473 3.711 12.915	3.239 x 10 <sup>9</sup> -- --	.070 .067 .076	.381 .340 .405	Realized we did not fully link -O2 correctly and fixed the error.
Midmark adventure Sandmark	0.460 3.267 13.095	2.886 x 10 <sup>9</sup> -- --	.068 .059 .077	.973 .880 1.014	Removed the stack data structure used for the unmapped segments
Midmark adventure Sandmark	0.393 2.438 10.900	3.007 x 10 <sup>9</sup> -- --	.058 .044 .064	.854 .746 .832	Condensed code. Got rid of .c and .h files for Handler and Segment. Now we just have um.c, execute.c, execute.h