Dylan Chelo & Nick Powers 5/10/23

Lab Notes

For the benchmarks for this assignment we are using a big benchmark of sandmark, a medium benchmark of a partial solution to the adventure game and a small benchmark of midmark. For the first 2 benchmark times, our load_prog was so unbelievably inefficient that we ran it for around 40 minutes and did not see any text pop up or any progress made so we put NA as we did not want to run it for longer since we needed to perform benchmark testing.

Benchmark	Time(sec)	Instructions	Rel to start	Rel to prev	Improvement
big medium small	487.005 NA 19.118	- - 482,601,705,512	1.000 1.000 1.000	1.000 1.000 1.000	No improvement (Starting point)
big medium small	463.220 NA 17.952	- - *482,601,705,512	0.951 NA 0.939	0.951 NA 0.939	Compiled with link-time-optimization turned on *We did not count the instructions again at this stage, it took over an hour and the change in time was minimal to none
big medium small	156.106 59.046 6.171	- - 48,288,402,483	0.321 1.00 0.323	0.337 1.00 0.344	Changed our load_prog function to do a mem swap instead of using a clone call (Clone was taking 40% of the execution time) *We counted this as the start for Medium
big medium small	91.683 30.908 3.186	- - 19,423,135,979	0.188 0.523 0.167	0.587 0.523 0.516	Replaced unnecessary get calls by creating a variable at the start of the loop and then using the variable instead, reduced time spent in get functions
big medium small	91.315 30.848 3.114	- - 19,424,481,340	0.188 0.522 0.163	0.996 0.998 0.977	Added inline tags above some of the opcode functions.

Benchmark	Time(sec)	Instructions	Rel to start	Rel to prev	Improvement
big medium small	18.978 6.063 0.468	- - 3,346,369,159	0.004 0.103 0.002	0.208 0.197 0.150	Changed our memory manager from a HashMap<u32> to a Vec<u32<u32>>, as well as changed our functions to reflect that. It was spending too much time trying to get values (approx. 67.2%)
big medium small	17.463 4.456 0.468	- - 3,346,369,159	0.036 0.075 0.002	0.920 0.735 1.00	Removed the Opcode enum and replaced it with hardcoded numbers.
big medium small	14.527 5.103 0.515	- - 3,395,894,075	0. 0. 0.	0. 0. 0.	Preallocates memory for the segment Vec<u32> to reduce the time spent allocating more memory to it. Introduces some overhead for the smaller data.