Tiffany Wong
CS351
Lab 4 Report

A separate (typed) design document (named lab4-report.pdf) describing the results in a table format. You must evaluate the performance of the various parameters outlined and fill in the table specified to showcase the results. You must summarize your findings and explain why you achieve the performance you achieve, and how the results compare between the various approaches. Highlight the best one in green, and worst one in red. Can you explain in words why they are the best, or the worst, and why the data you have makes sense.

method	num_ops	+ ops/sec	- ops/sec	* ops/sec	/ ops/sec
function	1 billion	70786056.505961	63122119.777126	63067501.083342	58809854.390917
pipe	1 million	20523.142285	18776.583535	20023.843993	20227.800227
socket	1 million	13272.376317	13013.890897	13821.648298	13762.032386

Overall, for my code, the function method performs the best and has the most operations per second. The socket method is the slowest, meaning it's the worst-performing method. Pipes are a unidirectional IPC mechanism, where one end of a pipe is opened for reading and the other end is opened for writing. Sockets can also connect two nodes on a network to communicate with one another, where one socket listens on a particular port at an IP, while other socket reaches out to the other to form a connection. In my data, I can see that the pipe method is faster than the socket method, and that's because pipes are implemented in memory on a single host where both communicating processes run. But ultimately what trumps both methods is the function method, which is the fastest because pipes have default behaviors. For example, the writing and reading ends of a pipe typically have blocking behavior, therefore slowing down the process, while the function method is straightforward and gets the job done quickly.