

My client.cpp was created using a model of request threads, Producer/consumer buffer, and worker threads. After starting the client and data server, the producer/consumer buffer is created using a standard semaphore implementation. Next, the request threads are created, one for each of the specified number of patients in the problem. In this case there were 3 patients that required data requests. Next, the proper number of worker threads are created, according to the input value of the program. 3 statistic threads were also created to keep track of the number of requests per each patient. All of these threads were created using the pthread_create() function, by passing in the required data for the helper function that was linked to the corresponding thread type. Sometime in the creation of the worker threads, there is a segmentation fault that causes the program to seize. This causes me to not be able to accrue data on the various inputs of the function. Although, I can make an educated guess at the results of the function depending on the inputs. Clearly, increasing the number of request threads will increase the computation time of the program. I think that increasing the number of working threads will have somewhat of a performance improvement, but would be fairly negligible.