

# Multithreaded File Processor Report

Muhammad Arsalan Hussain

December 3, 2023

## Introduction

This report discusses the implementation of a dynamic data structures in the provided C code, as well as the performance difference between single threaded and multithreaded programs.

## Dynamic Data Structure

The data structure used to store the input data is a dynamically allocated integer array. The array is initially set to `NULL`, and memory is allocated dynamically using `realloc` as new elements are read from the input file. This allows the data structure to adapt to the number of elements in the file without the need for a fixed-size array.

## Threaded Computations

The program employs multiple threads to concurrently compute the sum, minimum, and maximum values of the data. Each thread is responsible for a specific range of elements in the array. The dynamic allocation of thread-specific data parameters ensures that each thread operates on a distinct portion of the array.

## Mutex Usage

It's worth noting that mutex locks are used sparingly, only where necessary to maintain data integrity. In this case, the critical sections are small and infrequent, minimizing the impact on performance.

## Performance metrics

Time is measured in seconds for all computations. Values might be different during runtime as these are averaged values. Data Large was not properly tested due to hardware limitations. My machine only supported upto 4 threads so higher threads only made the performance worse hence it wasn't tested. But within 4 threads, the performance is far better as we can see in the table.

Table 1: Time comparison

<b>Number of threads</b>	<b>Data tiny</b>	<b>Data small</b>	<b>Data medium</b>
<b>1</b>	0s	0.49s	1s
<b>2</b>	0s	0.49s	1s
<b>4</b>	0s	0.49s	1s
<b>8</b>	0s	0.49s	1s