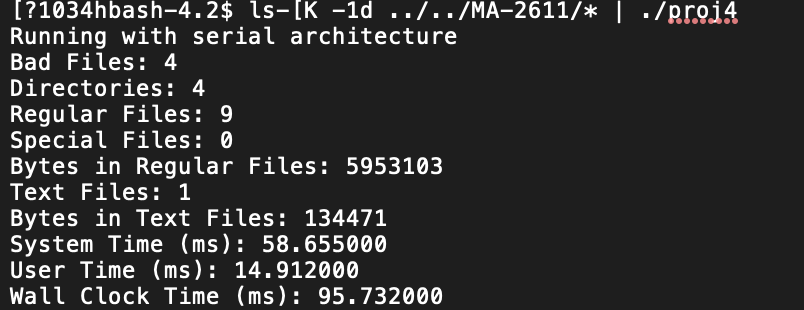
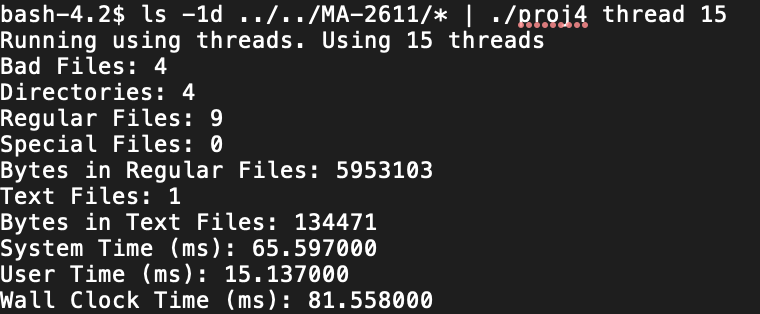
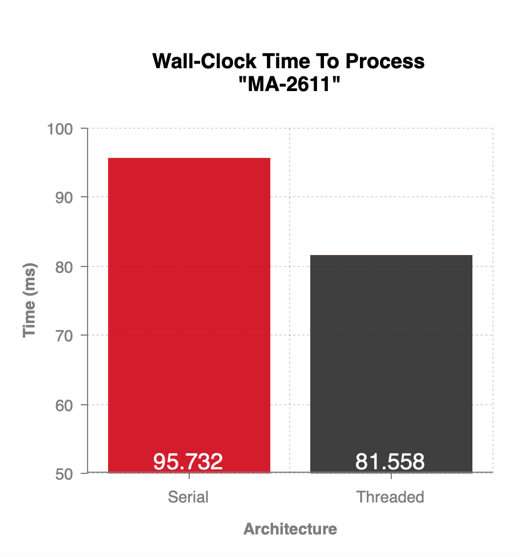
Time Analysis on Project 4

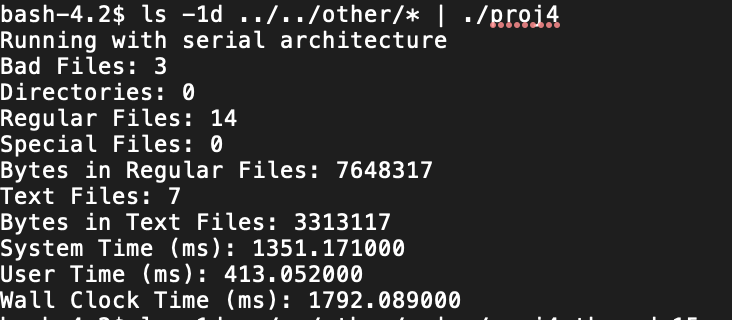
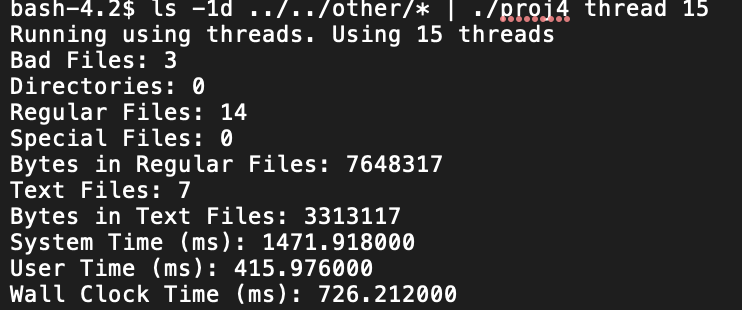
Harrison Burack

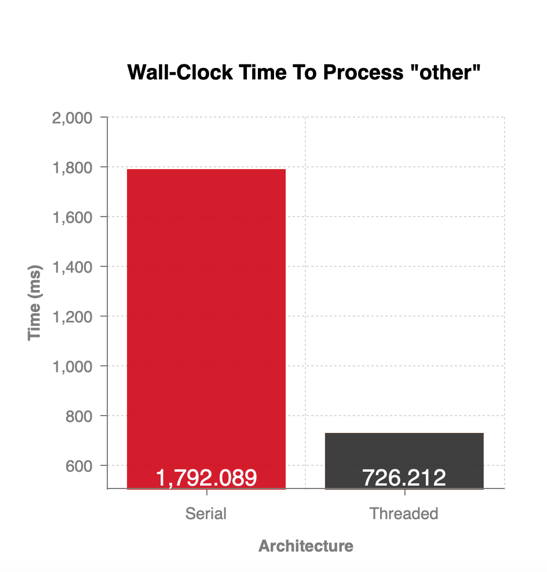
This time analysis is to determine the different in how long it takes a process to complete through a serial architecture compared to a threaded architecture. The time for these processes to complete is taken in milliseconds (ms). For this test, I used two directories that I already had created and saved on my storage drive *storage.wpi.edu*. The first directory was *MA-2611*, which contained 17 files. 4 of these files were classified as “bad” files, 4 were directories, and the remaining 9 were regular files. Of those 9 regular files, 1 file was a text file. The statistics for *MA-2611* are below, as well as a graph showing the wall-clock time performance of this directory being processed in the serial and threaded architectures.



The second directory I tested with was *other*, another directory from *storage.wpi.edu*. This directory also contained 17 files. 3 of those files were “bad” files, 0 were directories, and 14 were regular files. Of those 14 regular files, 7 were text files. The statistics for *other* are below, as well as a graph showing the wall-clock time performance of this directory being processed in the serial and threaded architectures.



From the statistics and graphs of the processes being completed in the serial and threaded architectures, it is clear that the threaded architecture is faster in terms of wall-clock time. This is due to the nature of threads allowing multiple processes to be done at the same time on multiple processors. When analyzing *MA-2611*, the threaded process is only shorter by a small amount because there is not much of a difference between the files. Most files are either regular or bad, and most of the calculation comes from determining the bytes of a text file. The performance of the threaded architecture is clearly better when analyzing *other*. Since there are more text files, it is clearer to see the difference in performance. Since the threaded architecture would be using more processors and scheduling each process accordingly, the threaded architecture can complete the process much quicker.