EECS 678
Stephen Longofono
Lab 8

1. The time required to copy the file using read_write varies with the size of the buffer specified. Smaller buffer sizes take longer. The time required for memmap varies much less regardless of how you perform the copy. Discuss why this is, and in your discussion consider the influence of: (1) the overhead of the read() and write() system calls and (2) the number of times the data is copied under the two methods. This question is worth 20 points.

Read and write system calls require that a copy of all data be made into a buffer. That is, to read, the data is copied to the buffer, and then for write, it is copied out of the buffer to the final destination. Furthermore, this involves a context switch for each read and write call.

The small buffer size means more iterations of reading to the buffer and writing to output. As long as the buffer size is small, this overhead time will exceed the time required for the copying of data, which will cause the process to be slow.

In contrast, mmap copies a part of the input into memory, giving direct access within memory. The output file is also mapped to memory. This allows a page at a time to be written at memory speeds (as opposed to disk speeds).

2. When you use the read_write command as supplied, the size of the copied file varies with the size of the buffer specified. When you use the memmap command implemented the size of the source and destination files will match exactly. This is because there is a mistake in the read_write code. What is the mistake, and how can it be corrected?

Instead of writing exactly the amount read from the input file, we are writing in chunks the size of the buffer. Whenever the size of the file does not evenly divide by the buffer size there is excess which does not align with the buffer size. This excess, rather the difference between it and the buffer size, will be the difference in the output file size. Thus for smaller buffers, there is less discrepancy, and for larger buffers, there is more discrepancy.