

Lab 10 Report

Nicholas Jurden 2415098

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 `mmap` 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.
 - In most cases, `mmap` requires less time because the information is copied and written less. In sequential I/O, many calls to read and write are made. this makes for higher overhead because copying data is relatively expensive. `Mmap` functionality reduces the amount of times data is copied in a I/O by using virtual memory outside of the heap to copy data. `Mmap` can especially reduce overhead for multiple programs accessing common files because the virtual memory assigned to reproducing a file is available in a shared memory space, with the correct `mmap` configuration.
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 `mmap` 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?

- The mistake arises when you write out the entire size of the buffer each time, rather than taking into account the size of the input each time. The buffer size, at any denomination, on the last write will likely include unused space that would affect the size of the file. Correct the mistake by checking to see if the buffer is full and if not, making the write smaller.