Joseph Pennington (2912079)

EECS 678 Lab 11 Report

1. The time required to copy the file using the read_write method takes a large amount of time to complete because it is limited by the size of the buffer. If the buffer is small, more read() and write() system calls would be needed to complete the copying task. However, if there is a large buffer, then the read_write method will run faster, to an extent, because there will be less read() and write() system calls. Additionally, another reason why the read_write method takes a large amount of time is because it must essentially copy the data twice. It first copies the data when reading into the buffer and then when writing the data from the buffer, which is very inefficient.

   The time required to copy the file using the memmap method is much smaller because the data is only copied once. With this method, the data is directly copied from the kernel-space buffer memory to the user-space buffer.

2. The mistake in the read_write code is that we are using a set buffer size while reading in chunks of memory. This means that if the buffer is reading 5 bytes at a time from a file that is 12 bytes, this method will then read an additional 5 bytes to account for the remaining 2 bytes. Therefore, the copied file may not be the same size as the input file. In general, the larger the buffer size, the larger the difference in file sizes may be. This mistake can be corrected by checking if the end of the input file has been reached. This solution can be implemented with the read() system call because it will return 0 when the end of the file is reached.