Project 2 Design Documentation

Goals:

Producer consumer problem. How can we speed up reading a file into a buffer, while only having a limited space.

Outline:

Set up two threads. One will handle the reading, one will handle the writing.

Set up three semaphores. One semaphore will keep track of the size of the buffer and if it is full. It will help read the file into the buffer. Another semaphore will keep track of reading the buffer into the output file. It will make sure there are items still in the buffer. The third semaphore will act as a mutex to make sure the threads aren’t competing over the same memory.

Solution:

My view on the matter was to set up the buffer as a queue. A queue works because it can only be accessed at the front, which is what we care about when reading. I used a while loop to make sure that the thread kept running. The semaphores made sure the buffer was kept intact.

Explanation:

With this set up there was no way that the threads could overlap on each other which safely transferred the data from one file to the buffer. And from the buffer to the output file.

Flow:

Thread 1{

If the semaphore is open (meaning there is space inside buffer)

input- > buffer : input ->buffer: input -> buffer …

}

Thread 2{

If the semaphore is open (meaning that there are items inside buffer)

buffer ->output : buffer -> output : buffer -> output …

}

File empty finishes