Rizwan Chaudry 4/20/14

PA5 – Multi-threaded book order system

In main I create my producer thread which reads in the files passed in by the user and then creates an instance of the order struct to hold the orders in the orders text file. My producer thread then creates a consumer thread which gets a mutex lock which allows it to apply the appropriate manipulations to the data (i.e. update credit, insert data into proper queue etc...). Thread synchronization is handled by appropriately locking the consumer threads at the proper time allowing any given consumer thread at that time to work with the data and then unlocking those consumers when they are done with the data. This way I was able to avoid race conditions and deadlocks. Once all consumer threads have finished and control returns to the producer thread, the producer thread returns and I then simply iterate through the proper queues and print the order information. Then lastly I free all memory that was allocated to my various structs. My program also not only prints the information to the terminal screen but, also prints all information into a text file as well.