Name: Mubasher Mehnaz Begum

Course: Bridge (Master of Science in Computer Science)

Assignment: CST – 221 Monitors and Semaphores

Date: 06/14/2020

# Detailed description of the scenario emphasizing the need for a synchronization mechanism

Both semaphores and monitors are used to solve the critical section problem (as they allow processes to access the shared resources in mutual exclusion) and to achieve process synchronization in the multiprocessing environment.

In my implementation, I have a scenario where both threads are utilizing a shared resource which is a credit card for making two online transactions. Synchronization is necessary because each thread seeks to utilize the credit card and deduct a payment for an online purchase. If synchronization was not present, then the value of the credit card could be changed before a process exits its critical region.

# Explanation of the pros and cons comparing monitors and semaphores in this specific scenario

For this scenario both Monitor and Semaphore act exactly the same. Both lock out the other thread while the initial process finishes then sends a signal to unlock and allow the other thread access to the shared credit card. As an actual Monitor does not exist in C programming language, we have to use a Mutex to represent. This is a huge disadvantage, given that the materials we are supposed to be using do not even exist.

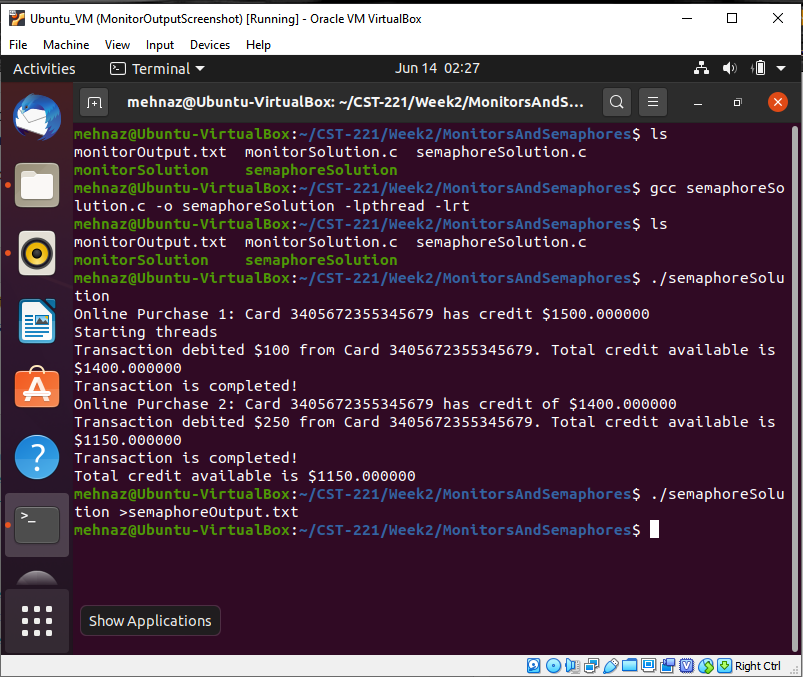
# A clear, justified recommendation, for the use of monitors or semaphores in this particular scenario

For this scenario it is simpler to use a Semaphore, both in implementation and code consistency. Ultimately both execute and output the same, so the main reason for Semaphore use is its ease. Instead of working with the locking mechanism, we can simply call the wait and signal functions.

# GIT Repository link:

<https://github.com/MubasherBegum/CST-221/tree/master/Week2/MonitorsAndSemaphores>

# Program execution results Screenshots: **Semaphores**



# Program execution results Screenshots: **Monitors**

