Tahsin Parvez
CSC 33200

Lab 6 Report

For the bank.c program, I initially defined all the variables, and then I defined the locking mechanism semMutex. For this entire process, there is both a locking mechanism and unlocking mechanism. For the first lock, I initiated this process by using P(semMutex), for the first child process with the father attempting to do an update. The dad will try to read the balance in the text file, and the following unlock signal in line 102. The unlock function is a signal written as V(semMutex). From the first lock to the first unlock, this is a critical section, as this will prevent race conditions from occurring. By doing so, the dad process is solely responsible for managing the balance(depositing, accessing ATM). The next lock is initiated at line 120, for when the son attempts to view the balance. Like any bank account, the son is able to withdraw money, only if the withdraw amount is less than that of the balance. The program will check if the balance is greater than the withdraw amount, and then after SON_1 writes the number of attempts remaining in the attempt.txt file, a signal to unlock is initiated. This is also a critical section because it will prevent any race conditions. During this process, only the son will be able to access the ATM. For the second child process, this is also done (SON_2). By unlocking and locking the process, this will prevent any issues because they will not be able to access the same processes at the same time. By doing so, race conditions will be prevented.