Question: Race conditions are possible in many computer systems. Consider a banking system that maintains an account balance with two functions: deposit (amount) and withdraw (amount). These two functions are passed the amount that is to be deposited or withdrawn from the bank account balance. Assume that a husband and wife share a bank account. Concurrently, the husband calls the withdraw() function and the wife calls deposit(). Write a short essay listing possible outcomes we could get and pointing out in which situations those outcomes are produced. Also, propose methods that the bank could apply to avoid unexpected results.

Answer:

Race condition xảy ra phương thức withdrawl() và phương thức deposit(). Vì mỗi người đều có quyền truy cập vào biến "Số dư hiện tại" (một biến được chia sẻ) và họ có thể giảm hoặc tăng nó.

Đây là một ví dụ về race condition: Số dư Hiện tại là $ 500. Chương trình của người chồng gọi withdrawl($100) và truy cập Số dư hiện tại là 500 đô la. Tuy nhiên, trước khi Số dư Hiện tại mới được lưu trữ, chương trình của người vợ gọi deposit($100) và truy cập Số dư hiện tại là 500 đô la. Sau đó, chương trình của người chồng thay đổi biến Số dư hiện tại được chia sẻ thành 400 đô la, tiếp theo là chương trình của người vợ thay đổi biến Số dư hiện tại thành 600 đô la. Với một lần withdrawl($100) và một lần deposit($100), Số dư hiện tại một lần nữa sẽ là 500 đô la, nhưng nó là 600 đô la.

Giải pháp là tạo một "Lock" trên biến Số dư Hiện tại. Theo thỏa thuận, mỗi phương thức phải "Lock" biến trước khi truy cập và thay đổi nó. Nếu một phương thức xác định rằng biến là "Locked", thì nó phải đợi trước khi truy cập nó.

The pseudocode:

```
method withdrawl(Amt)
    If BalanceLock is locked, then WAIT
    Lock (BalanceLock)
        Assign (Balance - Amt) to Balance
   UnLock (BalanceLock)
end withdrawl
```

```
method deposit(Amt)
    If BalanceLock is locked, then WAIT
    Lock (BalanceLock)
      Assign (Balance - Amt) to Balance
    Unlock (BalanceLock)
  end deposit
```