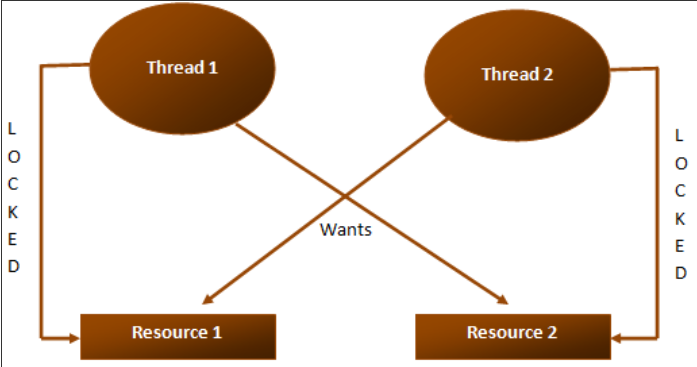
**Part 95 - Deadlock in a multithreaded program**

In this video we will discuss, why and how a deadlock can occur in multithreading with an example.

**Scenario when a deadlock can occur**  
**Let's say we have 2 threads**  
**a)** Thread 1  
**b)** Thread 2  
  
**and 2 resources**  
**a)** Resource 1  
**b)** Resource 2  
  
**Thread 1 has already acquired a lock on Resource 1 and wants to acquire a lock on Resource 2.** At the same time Thread 2 has already acquired a lock on Resource 2 and wants to acquire a lock on Resource 1. Two threads never give up their locks, hence a deadlock.



**Example code used in the demo**

usingSystem**;**

usingSystem.Threading**;**

publicclassProgram

**{**

publicstaticvoidMain**()**

**{**

Console.WriteLine**(**"Main Started"**);**

AccountaccountA=newAccount**(**101**,**5000**);**

AccountaccountB=newAccount**(**102**,**3000**);**

AccountManageraccountManagerA=new

AccountManager**(**accountA**,**accountB**,**1000**);**

ThreadT1=newThread**(**accountManagerA.Transfer**);**

T1.Name="T1"**;**

AccountManageraccountManagerB=new

AccountManager**(**accountB**,**accountA**,**2000**);**

ThreadT2=newThread**(**accountManagerB.Transfer**);**

T2.Name="T2"**;**

T1.Start**();**

T2.Start**();**

T1.Join**();**

T2.Join**();**

Console.WriteLine**(**"Main Completed"**);**

**}**

**}**

publicclassAccount

**{**

double\_balance**;**

int\_id**;**

publicAccount**(**intid**,**doublebalance**)**

**{**

this.\_id=id**;**

this.\_balance=balance**;**

**}**

publicintID

**{**

get

**{**

return\_id**;**

**}**

**}**

publicvoidWithdraw**(**doubleamount**)**

**{**

\_balance-=amount**;**

**}**

publicvoidDeposit**(**doubleamount**)**

**{**

\_balance+=amount**;**

**}**

**}**

publicclassAccountManager

**{**

Account\_fromAccount**;**

Account\_toAccount**;**

double\_amountToTransfer**;**

publicAccountManager**(**AccountfromAccount**,**

AccounttoAccount**,**doubleamountToTransfer**)**

**{**

this.\_fromAccount=fromAccount**;**

this.\_toAccount=toAccount**;**

this.\_amountToTransfer=amountToTransfer**;**

**}**

publicvoidTransfer**()**

**{**

Console.WriteLine**(**Thread.CurrentThread.Name

+" trying to acquire lock on "

+\_fromAccount.ID.ToString**());**

lock**(**\_fromAccount**)**

**{**

Console.WriteLine**(**Thread.CurrentThread.Name

+" acquired lock on "

+\_fromAccount.ID.ToString**());**

Console.WriteLine**(**Thread.CurrentThread.Name

+" suspended for 1 second"**);**

Thread.Sleep**(**1000**);**

Console.WriteLine**(**Thread.CurrentThread.Name+

" back in action and trying to acquire lock on "

+\_toAccount.ID.ToString**());**

lock**(**\_toAccount**)**

**{**

\_fromAccount.Withdraw**(**\_amountToTransfer**);**

\_toAccount.Deposit**(**\_amountToTransfer**);**

**}**

**}**

**}**

**}**