MultiThreading چیست ؟

مقدمه

گاهی اوقات ممکن است که شما بخواهید برنامه شما دو یا چند عمل را به طور همزمان انجام دهد و یا اینکه نیاز به انجام عملیاتی که مدت زمان زیادی به طول می انجامد و یا زمان انجام آن معلوم نیست ، باشد ، بدون اینکه برنامه شما از دسترس کاربر خارج شود و به اصطلاح برنامه شما تا پایان یافتن عملیات قفل کند و همچنین کاربر بتواند عملیات را متوقف/معلق/شروع دوباره نماید . در چنین موقعیتی نیاز به MultiThreading حس میشود . به فرض مثال کد زیر را در نظر بگیرید :

```
Integer = 0 To 10000000 For i As
For i2 As Integer = 0 To 100
Nothing Do'
Next
Next
```

هنگامی که این عملیات شروع میشود ، کاربر توانایی کار با برنامه تا پایان یافتن آن را نخواهد داشت .

Thread چیست؟

Thread نامی برای جریان اجرای یک عملیات خاص میباشد و هنگامی که برنامه شما دارای چند Thread میباشد بدان معناست که قسمت های مختلفی از کد برنامه شما به طور همزمان در حال اجرا شدن میباشند . در حقیقت کامپیوتر زمان پردازش یک عملیات را به قسمت(slice) های مختلفی تقسیم میکند و هنگامی که شما یک Thread جدید را آغاز میکنید کامپیوتر قسمتی از زمان را به آن اختصاص میدهد . لازم به ذکر است که برنامه شما از ابتدا دارای یک Thread اصلی (Main Thread) برای اجرا کد مربوط به آن میباشد .

کار خود با Thread ها را آغاز مینماییم :

میخواهیم برنامه ای بنویسیم که تا یک عدد معین عملیات شمارش را انجام دهد .

1 – **یک پروژه Windows Apploication به نام MutiThreading Sample ایجاد نمایید .**

2 – **یک Button به نام btnStart و یک TextBox به نام txtMAX به فرم اضافه نمایید .**

3 – **یک کلاس به نام clsCounter به پروژه اضافه کرده و کد زیر را در داخل آن قرارهید :**

```
clsCounter Public Class
Public MAX As Integer
(Number As Integer Public Event CountingFinished(ByVal

()Sub StartCounting

Dim intTotal As Integer
Integer = 0 To MAX For i As
intTotal += 1
```

```
Next
(CountingFinished(intTotal RaiseEvent
End Sub
End Class
```

توضیحات در مورد کد فوق :

- **وظیفه این کلاس شمردن از 1 تا مقدار MAX میباشد .**
- **رویدادی با نام CountingFinished تعریف کردیم که هنگامی که عملیات شمارش به پایان برسد اتفاق می افتد .**
- **متد StartCounting از 1 تا مقدار intMax را شماره کرده و در هر بار اجرای حلقه یک واحد به مقدار متغیر intTotal اضافه میشود که در نهایت مساوی با مقدار MAXخواهد بود .**
- **پس از پایان شمارش رویداد CountingFinished را همراه با پاس کردن متغیر intTotal به آن اجرا مینماییم .**

حال ما باید در هنگامی که دکمه کلیک میشود یک Thread جدید ایجاد کرده و سپس متد StartCounting کلاس clsCounter را اجرا کرده و رخدادن رویداد CountingFinished را کنترل نماییم . در زمانی که عملیات شمارش انجام میشود ما میتوانیم رابط کاربری را کنترل کرده و کاربر توانایی کار با برنامه را دارد .

حال کد زیر را به پروژه خود اضافه نمایید :

```
Sub CountingFinishedEventHandler(ByVal N As Integer(
System.Windows.Forms.MessageBox.Show("Counting Finished("!
End Sub
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnStart.Click
Dim CounterClass As New clsCounter
Dim CountingThread As New Threading.Thread(AddressOf CounterClass.StartCounting(
CounterClass.MAX = Val(txtMax.Text(
AddHandler CounterClass.CountingFinished, AddressOf CountingFinishedEventHandler
CountingThread.Start()
End Sub
```

توضیحات در مورد کد فوق :

- **ابتدا یک پروسیجر برای کنترل رویداد CountingFinished مربوط به کلاس Counter ایجاد مینماییم . هنگامی که رویداد اتفاق بیافتد(عملیات شمارش به پایان برسد) ، پیغامی مبنی بر پایان یافتن عملیات به کاربر نشان داده خواهد شد .**
- **در رویداد Click شی ء btnStart ، ابتدا یک نمونه از کلاس CounterClass ایجاد مینماییم .**

• سپس برای ایجاد شی ء Thread ، آدرس متد clsCounter.StartCounting را به سازنده کلاس Thread پاس مینماییم به طوری که متد clsCounter.StartCounting را بعد از آوردن کلمه کلیدی addressof ، می آوریم .

• بعد ، توسط کلمه کلیدی Addhandle ، کنترل کننده رویداد که CountingFinishedEventHandler نام دارد را به رویداد clsCounter.CountingFinished متصل مینماییم .

• در آخر نیز توسط متد Start مربوط به شیء CountingThread ، عملیات را آغاز مینماییم .

برخی متدهای دیگر مربوط به شی ء Thread :

Suspend و Resume : در حالی که یک Thread در حال اجراست ، توسط متد Suspend میتوانید آن را معلق کنید که منجر به متوقف شدن آن تا زمانی که متد Resume اجرا شود ، خواهد گردید .

Abort : Thread را متوقف میکند .

Sleep : توسط این متد میتوانید اجرای Thread را برای پاره ای از زمان (برحسب میلی ثانیه) به حالت تعلیق دربیاورید .

اولویت بندی Thread ها :

شما کنترل بیشتری بر روی Threadها دارید و میتوانید مقدار زمانی که هر Thread نسبت به دیگر Thread ها دریافت میکند را از طریق خاصیت Priority تنظیم نمایید . این خاصیت توسط یکی از ثابت های شمارشی زیر که عضوی از ThreadPriority میباشد تنظیم میشود :

ThreadPriority.AboveNormal : اولویت بالاتری به Thread میدهد .

ThreadPriority.LowerPriority : اولویت پایین تری به Thread میدهد .

ThreadPriority.HighestPriority : بالاترین اولویت را به Thread میدهد .

ThreadPriority.LowestPriority : پایین ترین اولویت را به Thread میدهد .

ThreadPriority.Normal : تولویت نرمال را به Thread میدهد .

پیدا کردن وضعیت Thread :

وضعیت یک Thread را میتوانیم به وسیله خاصیت ThreadState به دست بیاوریم که به وسیله یکی از ثابتهای شمارشی System.Threading.ThreadState معین میگردد .

System.Threading.ThreadState.Initialized : بیان میکند که Thread مقداردهی اولیه شده اما هنوز شروع نگردیده است .<Thread : System.Threading.ThreadState.Ready آماده است .

System.Threading.ThreadState.Running : بیان میکند که Thread در حال اجرا است .

System.Threading.ThreadState.Standbye : بیان میکند که Thread در حالت آماده به کاراست .

System.Threading.ThreadState.Initialized :بیان میکند که Thread به پایان رسیده است .

System.Threading.ThreadState.Transition : بیان میکند که Thread بین دو وضعیت بوده و در حالت انتقال از وضعیتی به وضعیت دیگر است .

System.Threading.ThreadState.Unknown : بیا میکند که وضعیت Thread معلوم نیست .

System.Threading.ThreadState.Wait : بیان میکند که Thread در حالت انتظار است .

مثال :

اگر چنین کدی داشته باشیم:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
Dim Thread1 As New Threading.Thread(AddressOf test(

Thread1.Start()


Dim Thread2 As New Threading.Thread(AddressOf test(

Thread2.Start()

End Sub
Public Sub test()

MsgBox("test("

End Sub
```

هر دو messageBox رو با هم می بینیم. اما اگر کد زیر رو بنویسیم:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
test()

test()


End Sub
Public Sub test()

MsgBox("test("

End Sub
```

ابتدا messageBox اول و پس از ok کردن دومی را خواهیم دید.

نویسنده : علیرضا مداح

منبع : www.barnamenevis.com