



منتدى الهندسة المعلوماتية
سوريا
حلب

كيفية انشاء كائن من نوع TThread في Delphi7

محمد سامر سروجي

الفهرس

- 3..... ما هو كائن TThread ؟
- 3..... كيفية التعامل مع TThread ؟
- 4..... ملاحظات على الكود :
- 7..... ملاحظات :

كيفية انشاء كائن من نوع TThread في Delphi7

هل واجهتك مشكلة تنفيذ أكثر من عملية في نفس الوقت؟؟ هل سمعت بمفهوم Thread (المسرب)؟؟ هذه المقالة تشرح مبادئ عمل ال Thread (المسرب) و كيفية كتابة برنامج يحتوي كائن من نوع TThread و ذلك بشرح مفصل و من البداية وذلك بإنشاء مشروع عملي و بعد الانتهاء من قراءتها ستكون قادراً على استخدام مفهوم Thread (المسرب) ضمن أي برنامج من برامجك .

المشروع هو مجرد مشروع بسيط بأن تقوم أكثر من Label بزيادة محتواها بشكل تدريجي و في نفس الوقت. لا يمكن كتابة هذا التطبيق بدون استخدام Thread (المسرب) لأنه إذا كتبنا

```
For i:=1 to 1000000 do
Begin
Label1.Caption:=inttostr(i);
Sleep(10);
End
```

من أجل أول Label و بالمثل للباقي فلن تبدأ Label2 مثلاً بالعد الا عند انتهاء Label1 و لكننا نريد من كل Label أن تعد معا فما هو الحل؟؟؟

الحل هو بإنشاء كائن TThread يقوم بعملية العد ضمن Label بحيث نشأ Thread (المسرب) من أجل كل Label.

ما هو كائن TThread ؟

ببساطة اذهب على قائمة File في Delphi7 ثم اختر New و من ثم Other ثم اختر من الواجهة التي تظهر لك ايقونة Thread Object

ثم قم باعطائه اسم مثلاً TCountThread و عندها و ببساطة تكون قد أنشأت TThread جديد.

كيفية التعامل مع TThread ؟

سوف يظهر لك Unit جديدة تحوي صنف Type فيه

```
TCountThread = class (TThread)
```

بالإضافة إلى Procedure اسمه Execute وضمنه يوجد تعليق *{ Place thread code here }* أي أننا سنكتب الكود في هذه المنطقة و عند بناء غرض من TCountThread سيقوم بتنفيذ هذا الاجراء و ذلك بفتح Thread (المسرب) جديد خصيصاً لهذا الاجراء Execute.

قبل أن أكمل في شرح TCountThread لنعد إلى مشروعنا كنا نريد أن نجعل عدة Label تعمل معاً لذلك نعود للتطبيق Unit1 و نضع على الشكل Form1 كل من العناصر التالية :

Label1, Label2, Button1

و قبل كل شيء نقوم باستخدام الوحدة التي تتضمن الكائن TCountThread و لنعرف هنا متحول عام اسمه First و ذلك بكتابة قبل كلمة implementation :

```
Var First : Boolean;
```

نكتب ضمن Button1 الكود التالي:

```
var Counter:TCountThread;
begin
  Counter:=TCountThread.Create(False);
end;
```

هذا يعني أنه عند الضغط على الزر فسوف يتم انشاء Thread (المسرب) جديد اسمه Counter وسيبدأ بتنفيذ التابع Execute (لأن القيمة الممررة للباني هي False أما إذا كانت True فلن يبدأ بتنفيذ التابع Execute).

و نستطيع مقاطعة عمل Thread و ايقافه باستخدام التابع Suspend و اذا اردنا المتابعة نستخدم التابع Resume

لنعد إلى الكائن TCountThread و لنستخدم Unit1 ضمن هذه الوحدة (هذا لا يفضل و لكن مرحليا سنتجاوز ذلك)

نكتب ضمن التابع Execute الكود التالي

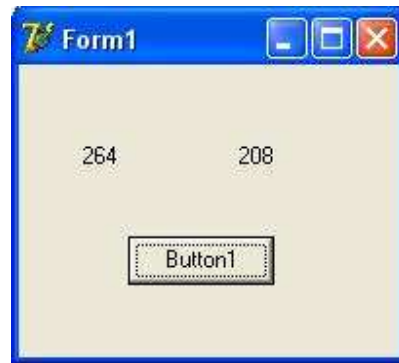
```
procedure TCountThread.Execute;
var i:integer;
begin
  First := not First;
  if First then
    for i:=1 to 1000000 do
      begin
        Form1.Label1.Caption:=inttostr(i);
        sleep(20)
      end
    else
      for i:=1 to 1000000 do
        begin
          Form1.Label2.Caption:=inttostr(i);
          sleep(20)
        end;
      end;
end;
```

ملاحظات على الكود :

يجب استخدام الوحدة SysUtils في Unit2 التي تتضمن TCountThread من اجل أن تتعرف على التابع inttostr.

في حال عدم كتابة التابع Sleep أي عدم استخدام أي تأخير نهائيا نجد تقطع في التبادل بين الكتابة على Label1 و الكتابة على Label2 لأنه لن يجد وقتا للتبديل و لكن باستخدام Sleep و لو حتى Sleep(1) فسنرى غياب هذا التقطع و التعليل واضح.

و الآن ننفذ برنامجنا الفذ نلاحظ أنه عند الضغط على الزر لأول مرة سيتم بدأ عمل العد لـ Label1 و عند الضغط على الزر مرة أخرى سيتم بدأ عمل العد لـ Label2 و سيعملان معاً



نلاحظ أن هذا الكائن TCountThread مرتبط مع Unit1 و مع Label1 و Label2 بالتحديد لذلك يجب الفصل بينهما حتى يصبح الكائن TCountThread غير مرتبط ب Label معينة و بالتالي يجب تمرير Label كوسيط له .

قبل ذلك لنشرح أهم التوابيع المستخدمة في الكائنات من النوع TThread و بفرض لدينا الغرض :

Counter: TCountThread;

الذي صنعناه في المحاضرة السابقة :

1-TCountThread.Create(CreateSuspended:Boolean) ;

يقوم هذا التابع بإنشاء غرض من النوع TCountThread و إذا كانت قيمة الوسيط الممرر هي True فإن Thread (المسرب) المنشأ لن يبدأ بالعمل مباشرة أما إذا كانت القيمة هي False فإن Thread (المسرب) المنشأ سيبدأ بالعمل مباشرة. false لأن المتحول الذي نمرر له هذه القيمة اسمه CreateSuspended أي هل تريد الخلق بدون أنا يبدأ Thread (المسرب) بالعمل فإذا مررنا True فنحتاج بعدها لكي يعمل Thread (المسرب) باستدعاء الاجراء Resume ليبدأ بالعمل

و هنا نجد التابع الثاني :

2- Counter.Resume

ويقوم هذا الاجراء بمتابعة عمل ال Thread (المسرب) و ذلك إما إذا خلقناه في حالة توقف او اذا قمنا بايقاف عمله خلال البرنامج

و من هنا قد نحتاج لتابع يوقف ال Thread (المسرب) و هو

3- Counter.suspend

ويقوم هذا الاجراء ايقاف عمل ال Thread (المسرب) و ذلك دون قتل ال Thread (المسرب) نهائيا و يمكننا امر Thread (المسرب) بمتابعة عمله كما ذكرنا سابقا باستخدام التابع Resume.

أما التابع

4- Counter. Terminate

فيقوم بقتل ال Thread (المسرب) نهائيا و تحريره من الذاكرة

و الآن لتتابع عملنا على المشروع كنا نريد فصل Thread (المسرب) عن Label معينة إذا لكل Thread (المسرب) سيحوي متحول من نوع TLabel وليكن Count:TLabel خاص به و بالتالي الحل الوحيد لدينا هو تمرير Label له كوسيط عند بدأ خلق Thread (المسرب) إذا سنحتاج لورثة التابع Create و التعديل في محتواه بحيث نمرر لل Thread (المسرب) من خلاله ال Label المراد زيادة محتواها .

و بالتالي نعرف في قسم Public للكائن TCountThread التابع الباني Create (ملاحظة كلمة Public لا تكون مكتوبة لذلك نضيفها نحن بعد Protected) على الشكل :

```
Public
  Constructor Create (Counter:TLabel;CreateSuspended:Boolean);
```

نلاحظ أننا نحتاج لإسناد الLabel التي أتت من هذا التابع إلى أخرى خاصة بالكائن TCountThread و بالتالي نعرف في قسم Private للكائن كل مما يلي :

```
Count : TLabel;
Procedure StartCounting;
```

و سنشرح وظيفة التابع StartCounting فيما بعد

و الآن نكتب في قسم implementation تعريف تابع Create كما يلي :

```
Constructor TCounterThread.Create (Counter:TLabel;CreateSuspended:Boolean);
begin
  inherited Create(CreateSuspended);
  Count := Counter;
end;
```

من الواضح أن التابع يقوم أولاً بتنفيذ التابع القديم من Create ثم يقوم بإسناد Label الممررة له إلى تلك الموجودة لديه Count

و الآن نكتب ضمن التابع Execute ما يلي :

```
procedure TCounterThread.Execute;
begin
  StartCounting;
end;
```

أي أن التابع Execute سيقوم بتنفيذ التابع StartCounting.

و من هنا يكون محتوى التابع StartCounting الذي قمنا بتعريفه في قسم Private هو :

```
procedure TCounterThread.StartCounting;
var i:integer;
begin
  for i:=0 to 1000 do
  begin
    Count.Caption:=inttostr(i);
    sleep(10);
  end;
end;
```

و هكذا نكون قد أكملنا انشاء هذا الكائن و بقي التطبيق.

ملاحظات :

يجب استخدام الوحدات التالية ضمن الكائن StdCtrls, SysUtils واحدة ليتعرف على TLabel و الأخرى ليتعرف على التابع IntToStr

بعد كتابة ترويسة التابع مثل Procedure StartCounting ضمن أحد أقسام الكائن (Public, Protected, Private) قم بضغط المفاتيح (Ctrl+Shift+C) ليقوم Delphi ببناء جسم التابع فإذا لم يكن جسم التابع قد تم بناؤه بعد سيعطي الخطأ التالي (حتى و لو كان تعريف الترويسة خالي من الأخطاء) :

Unsatisfied forward or external declaration: 'TCounterThread.StartCounting'

يفضل عدم كتابة الكود ضمن التابع Execute مباشرة و لكن كما فعلنا في هذا الكائن كتابة الكود في إجراء آخر و استدعاؤه ضمن التابع Execute

و الآن نعود للبرنامج التطبيقي



أعتقد أن الكود أصبح واضحاً :

في البداية نقوم باستخدام الوحدة التي تتضمن الكائن TCounterThread و ثم في قسم implementation و بعد استخدام الوحدة السابقة نعرف المتحولين :

```
var C1,C2: TCounterThread;
```

و الآن ضمن الأزرار نكتب الكود التالي :

```
procedure TForm1.Button_Create1Click(Sender: TObject);
begin
  c1:=TCounterThread.Create(Label1,false);
end;

procedure TForm1.Button_Create2Click(Sender: TObject);
begin
  c2:=TCounterThread.Create(Label2,false);
end;

procedure TForm1.Button_Suspend1Click(Sender: TObject);
begin
```

```
c1.Suspend;  
end;  
  
procedure TForm1.Button_Suspend2Click(Sender: TObject);  
begin  
    c2.Suspend;  
end;  
  
procedure TForm1.Button_Resume2Click(Sender: TObject);  
begin  
    c2.Resume;  
end;  
  
procedure TForm1.Button_Resume1Click(Sender: TObject);  
begin  
    c1.Resume;  
end;
```

و أعتقد أنه من الواضح من اسم كل زر و محتواه ما هي العملية التي يقوم بها (ما بدھا شرح صرتو أذكياء كفاية لفھمھا)

ملاحظة : يمكن بضغط الأزرار (Ctrl+Alt+T) مشاهدة Threads التي لدي و كيف تقتل نفسها عندما تنتهي من العمل .

و هكذا تكون المقالة قد انتهت أرجو ان تنال هذه المقالة اعجابكم .

و بانتظار آرائكم و ملاحظاتكم ضمن المنتدى على الرابط :

<http://www.alepposoft.com/info/showthread.php?t=299>

حلب

السبت، 19 آذار، 2005