

# Threads

22

Programación II y Laboratorio de Computación II

Edición 2018

# Hilos de Ejecución

- En sistemas operativos, un **hilo de ejecución, hebra o subproceso** es una secuencia de tareas encadenadas muy pequeña que puede ser ejecutada por un sistema operativo.
- Un hilo es simplemente una tarea que puede ser ejecutada al mismo tiempo que otra tarea.
- Los hilos de ejecución que comparten los mismos recursos, sumados a estos recursos, son en conjunto conocidos como un proceso.
- Un hilo secundario va a tener su propia pila de ejecución, independiente de la del hilo principal.

# Hilos de Ejecución

- El proceso sigue en ejecución mientras al menos uno de sus hilos de ejecución siga activo.
- En el momento en el que todos los hilos de ejecución finalizan, el proceso no existe más y todos sus recursos son liberados.





# Ejemplo básico

```
// Agrego la biblioteca
using System.Threading;

// ...

// Creo el hilo
Thread t = new Thread(UnMetodo);
// Inicio el Hilo
t.Start();
```

# Hilos Parametrizados

- El método utilizado puede tener parámetros.
- Para esto deberemos utilizar `ParameterizedThreadStart` al instanciar el nuevo hilo.
- El parámetro se pasará mediante el método `Start` de dicho hilo.

# Ejemplo con parámetros

```
// Creo el hilo  
Thread t = new Thread(new  
ParameterizedThreadStart(Metodo));  
// Inicio el Hilo, indicando el parámetro  
t.Start(1000);
```

```
private void Metodo(object o)  
{  
    Console.WriteLine((int)o);  
}
```



# Hilos y Controles Visuales

- Si deseamos modificar un control visual de un formulario (TextBox, ComboBox, Label, etc.) desde un hilo diferente al principal (“dueño” de estos controles) deberemos invocar a dicho hilo.
- Para esto le consultaremos al control si necesita ser invocado el hilo principal (InvokeRequired).
- Luego invocaremos dicho hilo (BeginInvoke) mediante un delegado.

# Ejemplo con controles de Form

```
private void Metodo(object o)
{
    if (this.label.InvokeRequired)
    {
        this.label.BeginInvoke((MethodInvoker)delegate()
        {
            this.label.Text = (int)o;
        });
    }
    else
    {
        this.label.Text = (int)o;
    }
}
```



# Hilos y Controles Visuales

- Dicha invocación puede necesitar parámetros.
- Para resolver este caso, utilizaremos un array de Object.
- Al realizar el Invoke (sincrónico, espera que un Thread finalice para ejecutar otro) o BeginInvoke (asincrónico) se pasará el delegado y dicho array.

# Ejemplo con controles de Form

```
delegate void Callback(string t);  
public void OtroMetodo(string texto) {  
    if (this.textBox.InvokeRequired) {  
        Callback d = new Callback(this.OtroMetodo);  
        object[] objs = new object[] { texto };  
        this.Invoke(d, objs);  
    } else {  
        this.textBox.Text = texto;  
    }  
}
```