COMPUTING SUBJECT:	**Socket programming**
TYPE:	Assignment
IDENTIFICATION:	SocketConcurrent
COPYRIGHT:	*Michael Claudius*
LEVEL:	Intermediate
TIME CONSUMPTION:	1 hours
EXTENT:	50 lines
OBJECTIVE:	TCP-sockets concurrent style
PRECONDITIONS:	Computer Networks Ch. 2.7
COMMANDS:	

IDENTIFICATION: SocketConcurrentC#

The Mission

We are going to explore the TCP socket programming using a concurrent server.

Precondition

You have done the assignment SocketIterative and have a running iterative solution

Useful C# links

- http://msdn.microsoft.com/en-us/library/5xt1dysy(v=vs.90).aspx
- http://blogs.msdn.com/b/pfxteam/archive/2010/06/13/10024153.aspx

Now we will restructure the program, SocketIterative, to utilize Thread and/or Task using the following classes.

TCPEchoServer1: A concurrent server class for management and setting up the server socket connection.
EchoService: A service Model class handling the connection socket communication and data manipulation.

Assignment 1. Application class: TCPEchoServer1

In *main* make the following changes:

Create a new Thread object myThread using service.doIt as a parameter
Call the myThread.start() method.

Compile and run!
Then start the TCPClient1 and see that everything operates fine….

Assignment 3. Several clients

Start the special version of TCPClient1 or TCPClient2 from several computers at the same time.
What happens? Compare it with what happened before when using an iterative server.

Assignment 4. Task using Thread pool

Instead of using a Thread class directly a Task/ThreadPool is advantageous.
Investigate how to apply a cashed thread pool by using the Task.Factory…...in main.

Assignment 5. Task vs Thread

What is the difference between a Thread, a ThreadPool and a Task.