**COMPUTING SUBJECT: Http and socket programming**

**TYPE:** Assignment

**IDENTIFICATION:** HttpServerChallenge

**COPYRIGHT:** *Michael Claudius*

**LEVEL:** Difficult

**TIME CONSUMPTION:** 1-5 hours

**EXTENT:** 50 lines

**OBJECTIVE: TCP-sockets**

**PRECONDITIONS:** Computer Networks Ch. 2.7, HttpServerStart

**COMMANDS:**

**IDENTIFICATION:** HttpServerChallenge

The Mission

We are going to explore the TCP socket programming by setting up an EchoServer which returns the client-sentence capitalized.

Maybe a useful C# link:

* <http://stackoverflow.com/questions/365370/proper-way-to-stop-tcplistener>

Maybe not !

Precondition

You have done the assignment SocketHttpStart

*Assignment 1 Graceful shutdown*

Until now we've shutdown the HTTP server by typing Ctrl-C in the console window running the server or just killing the window. That is not a very nice way to end a program, especially not if the program needs to do some clean-up before closing.

First implement some simple ideas like:

1. Sendig a special stop command, what are defects of this idea?
2. Sending a special htpp request, what are defects of this idea?

Then implement the idea below

1. A graceful shutdown
2. Make the HTTP server spin of another thread ("the stopping thread"). This thread makes a server socket listening on some port (the shutdown port).
3. When another process connects to this port the stopping thread calls some method (like stop()) on the server thread to inform the server thread that it is time to shut down.

When you want to shut down your HTTP server you simply start another program (a shutdown client). This program connects to the shutdown port.

Maybe the C# keyword volatile might be handy.

The C# method Pending might be handy.