Documentation lab 4 (Futures and continuations) - Mos Daniele 935

Implementations:

1. Directly implement the parser on the callbacks (event-driven);
2. Wrap the connect/send/receive operations in tasks, with the callback setting the result of the task;
3. Like the previous, but also use the async/await mechanism.

Entities:

- Custom-Socket:
  - has a socket to connect to the server
  - BUFF_SIZE (buffer size) which is 1024 bytes
  - buffer
  - response content, as a StringBuilder
  - id - keep a unique id for each custom socket
  - hostname
  - endpoint
  - remoteEndPoint - ip of website endpoint (address+port)
- Parser:
  - returns request string

- returns content length

Callbacks Implementation:

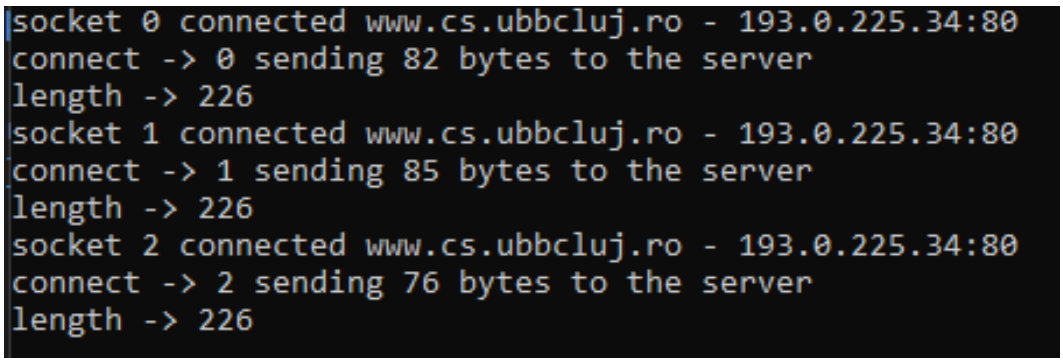
- run function gets activated, then it enters in the Start one providing the unique id and the host from the list as a string
- Start uses BeginConnect to connect to the remote endpoint
- uses Send function to send data to the server
- and uses BeginReceive to receive the server response data after sending
- Receiving function uses EndReceive to read response data
- check if the response header has not been fully obtained, get the next chunk of data

Test run

- using these urls:

"www.cs.ubbcluj.ro/~motogna/LFTC",
"www.cs.ubbcluj.ro/~rlupsa/edu/pdp/",
"www.cs.ubbcluj.ro/~forest"

- result:



Tasks Implementation:

- run function checks if we want it to run normally of async
- use tasks as a list of Task objects

```
var taskList = new List<Task>();
```

Normal:

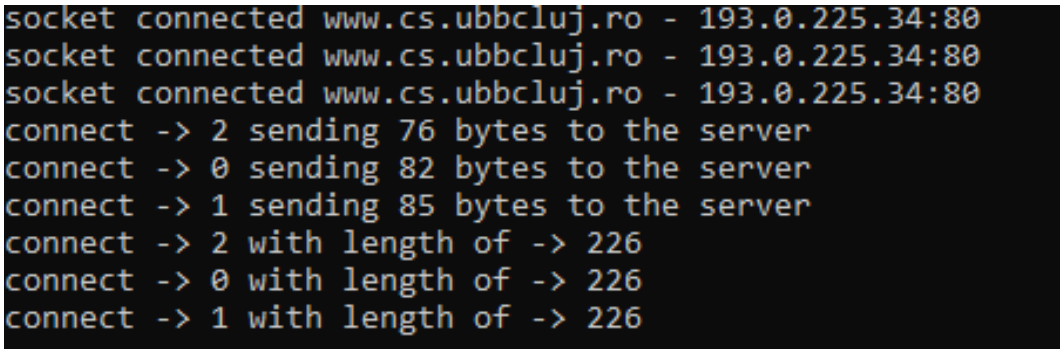
- Start connects to the remote server, requests data from server and receives server response using Connect/Send/Receive, then releases the socket
- Connect uses the BeginConnect function and it is blocked until signaled
- Send sedns data using BeginSend after converting the string data to byte data using ASCII encoding
- Receive uses BeginReceive to receive data

Test run

- using these urls:

"www.cs.ubbcluj.ro/~motogna/LFTC",
"www.cs.ubbcluj.ro/~rlupsa/edu/pdp/",
"www.cs.ubbcluj.ro/~forest"

- result:



Asynchronously:

- AsyncStart creates a TCP/IP client socket
- Using async functions ConnectAsync/SendAsync/ReceiveAsync for the task (all of them using await), after that we shut down the client
- ConnectAsync uses BeginConnect to connect and it's blocked until signaled
- SendAsync encodes the data to bytes using ASCII encoding and it sends data with BeginSend
- ReceiveAsync using BeginReceive to receive the data

Test run

- using these urls:

"www.cs.ubbcluj.ro/~motogna/LFTC",
"www.cs.ubbcluj.ro/~rlupsa/edu/pdp/",
"www.cs.ubbcluj.ro/~forest"

- result: