DT6

Arbeitsblatt: DNET2

Name: Manuel Thalmann Kurznamen: thalmma5

.NET Core

Aufgabe 1 Windows Client

Erstellen Sie einen Klienten für den beigefügte Server (Server5.c) als eine in C# .NET Core Kommandozeilen Applikation. Der Server ist in C geschrieben und lässt sich sowohl unter Windows als auch unter Linux kompilieren (mit gcc oder msc). Für Testzwecke können Sie auch den dazugehörigen in C geschriebenen Klienten (Client5.c) verwenden. Hinweis: es muss noch mit der ws2_32.lib Bibliothek gelinkt werden.

Die Anwendung soll die Daten asynchron zeilenweise empfangen und beim Empfang jeweils einen Event auslösen. Im Event werden dann die empfangenen Daten auf die Konsole geschrieben.

Hinweise:

- für die Abgabe muss mit 127.0.0.1 und Port 8000 gearbeitet werden
- Orientieren Sie sich an SocketListener.cs und SocketClient.cs

Abgabe:

Praktikum: DT6

Datei: DT6_solution.cs

Aufgabe 2 Linux Client

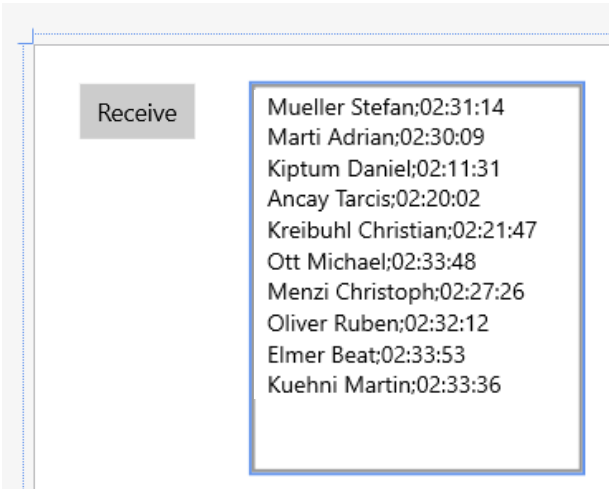
Dieselbe Anwendung soll unter Linux unter .NET Core zum Laufen gebracht werden. Installieren Sie dafür .NET 5 auf Linux und übersetzen und testen Sie das obige Programm.

Hinweis:

- Der Server soll/kann weiterhin unter Windows laufen.

Aufgabe 3. UWP Client

Es soll eine UWP Applikation erstellt werden, die die Daten via Socketverbindung empfängt. Das GUI enthält lediglich einen Knopf und eine mehrzeilige Textbox. Der Aufruf und der Update der Textbox soll asynchron erfolgen.



Hinweis:

- <TextBox x:Name="TextBox1" AcceptsReturn="True" ….
- Es kann wieder ein AsyncHelper erstellt werden

```
public static class AsyncHelper {
  public static async void InvokeIfRequired(this Page page, Action action) {
   await page.Dispatcher.RunAsync(Windows.UI.Core.CoreDispatcherPriority.High,
  () =>{action();});
  }
}
```

Aufgabe 4. App Installer und Boundle (optional)

Es soll ein App Installer und Boundle erstellt werden. Dafür muss wie in den Unterlagen beschrieben die App signiert werden.