

NVS Beispiel

Client/Server Beispiel mittels asio
Zeitsynchronisation nach dem Verfahren von Cristian

Client wie Server simulieren eine eigene Uhr (wie im Übungsbeispiel oder der praktischen Arbeit).

Die Uhr des Client lassen wir langsamer gehen als die Serveruhr (selber festlegen, aber so, dass dies sichtbar ist).

1. Der Client verbindet sich mit dem Server
2. synchronisiert danach drei Mal (in regelmäßigen Abständen) seine Zeit mit dem Server.

Die Ports werden als Option (default: 9999) den startenden Programmen über die Kommandozeile mitgegeben.

Der Client erhält als Positionsargument auch noch einen Clientnamen (z.B. "client1").

Zur Synchronisation sendet der Client eine Nachricht an den Server, die den Clientnamen enthält, der für die Ausgaben verwendet werden kann.

1. Der Server empfängt die Nachricht
2. wartet eine zufällige Zeit (1-3 Sekunden)
3. sendet danach diese gewartete Zeit an den Client
4. Der Client liest die Antwort und berechnet sich daraus die eigene Zeit, die auch in seiner Uhr gesetzt wird.

Nach dreimaliger Synchronisation beendet der Client die Verbindung durch Schließen und beendet sich danach selber.

Fehler sind abzufragen und zu behandeln.

Die Ausgaben erfolgen mittels =spdlog= und die Informationen werden Server wie Client mittels =CLI11= zur Verfügung gestellt.

Es können sich mehrere Clients gleichzeitig mit dem Server verbinden.

Dafür ist ein Repository mit Ihrem Namen anzulegen, das ein Meson-Projekt enthält. Für das Repo bekomme ich bitte einen Zugang. Entweder Mercurial oder git. Den Link schicken Sie mir bitte. Coding Conventions sind einzuhalten.

Checklist

- Klasse Uhr implementieren und testen
- Client Grundaufbau
- Server Grundaufbau
- Verbindung testen
- Zeit auf beiden ausgeben (Client langsamer als Server)
- Client sendet Anfrage mit Namen an Server -> Server gibt aus
- Client sendet Zeit an Server mit Namen
- Server bekommt Anfrage -> wartet 1-3 Sek und sendet Wartezeit zurück
- Client bekommt Zeit und berechnet neue Zeit
- Client prüft ob 3mal bekommen hat und schließt sich dann selbst