Testat: “Verteiltes Blackboard”

Thomas Vogt (thomas@thomas-vogt.de)

17. Mai 2022

1 Was ist ein Blackboard

Ein Blackboard dient dazu Informationen zu speichern. Es wird nur die zuletzt aktualisierte Information gesichert. Diese Informationen können in einem verteilten System von mehreren Empfängern gelesen und (in der Regel) von einem Sender aktualisiert werden

Nachrichten können veralten, das heißt sie werden nach einer definierten Zeit als ungültig markiert.

Nachrichten können aktualisiert werden; Alte Nachrichten werden bei der Aktualisierung überschrieben, die Gültigkeit zurückgesetzt (auf Gültig).

2 Aufgabenstellung

- Erstellung eines Servers, der Dienste zum Erstellen, Löschen von Blackboards, sowie zum Schreiben und Lesen von Nachrichten auf jenen “Blackboard”:
  - Mehrere Blackboard sollen über einen eindeutigen Namen identifiziert werden.
  - Der Server ist als Einzelinstanz ausreichend – Redundanz / Ortstransparenz/ Migrationstransparenz/ Relokationstransparenz/ Replikationstransparenz kann vernachlässigt werden
  - Nebenläufigkeitstransparenz soll beachtet werden, d.h. der Parallele Zugriff auf Ressourcen des Servers von mehreren Clients ist zu ermöglichen und sicherzustellen.
  - Die Art der Umsetzung (Sperren o.ä.) ist frei —Die konkrete Umsetzung ist zu dokumentieren.
  - Die Sichheitsanforderung sind reduziert: Verschlüsselung und Authentifizierung sind nicht notwendig zu berücksichtigen/ implementieren
    Ein Log soll geführt werden, dass alle Anfragen mit Quell-Addresse protokolliert

  - Auf Robustheit soll geachtet werden (z.B. falsche, ungenügend geprüfte Parameter auf Client-Seite)
  - Die Blackboard Namen sowie die Nachrichten können in ihrer maximalen Länge beschränkt werden. Mindestanforderung:
    Namen: 32 ASCII-Zeichen
    Nachrichten: 255 ASCII-Zeichen
  - Zeitangaben (Gültigkeit) können in Sekunden angegeben werden
  - Die Anzahl an zu erstellende Blackboards kann begrenzt werden. Mindestens 32 Blackboards sollen unterstützt werden.
  - Beschränkungen an zulässigen/ verwendeten Zeichensätzen ist zu dokumentieren.
  - Persistenz (Speicherung der Zustände über Neustarts des Servers hinaus ist nicht erforderlich.
  - Die Schnittstellen müssen nicht exakt wie vorgegeben (Rückgabeart, o.ä.) implementiert werden, jedoch sollen alle geforderten Anforderungen erfüllt werden.

- Erstellung eines Clients, der die Dienste des Blackboard Servers nutzt:
  - Die einzelnen Dienste sollen unabhängig voneinander genutz werden können, z.B. als Command Line Parameter:

    ```
    blackboard_client --server <address> --create MESSAGE_1
    blackboard_client --server <address> --write MESSAGE_1 MESSAGE_DATA_1
    blackboard_client --server <address> --read MESSAGE_1
    …
    ```

  - Antworten des Servers sollen dem Benutzer angezeigt (z.B. Logausgabe) werden.

- Die zu verwendende Programmierumgebung/ Technologie ist frei, ist aber mit dem Dozenten abzustimmen. (Unterstützte OSe Windows, Linux, Android; Programmiersprachen C, C++, Java,Python,.NET, PHP)

- Die Verwendung von Frameworks und Libraries ist erlaubt, aber zu dokumentieren.
  Eine eigene Leistung sollte ersichtlich sein

- Erstellung eine Dokumentation mit mindestens folgenden Inhalten:
  - Namen der Teammitglieder
  - Architekturbeschreibung mit Begründung der Entwurfsentscheidung
  - Anleitung zum Erstellen und Ausführen der Anwendungen
  - Dokumentation der Kommunikationsschnittstellen und Protokolle, Zulässige Parameter (Wertbereiche, Typen), Pre- und Post-Conditions, Fehlerverhalten

- Qualitätskriterien
  - Ausarbeitung/ Dokumentation entsprechend den Anforderungen
  - Lesbarkeit und Verständlichkeit des Codes
  - Funktionalität erfüllt
  - Quelltextdokumentation

3 Beschreibung der Dienste

3.1 CREATE_BLACKBOARD

Erstellt auf dem Server eine neues leeres Blackboard.

3.1.1 Parameter

Name Name des Blackboards

Gültigkeit Gültigkeit einer Nachricht in Sekunden (0 = unbegrenzte Gültigkeit)

3.1.2 Rückgabe

Erfolg oder Misserfolg:

- Blackboard erfolgreich erstellt
- Blackboard existiert schon
- Ungültige Parameter
- Weitere interne Fehler

3.2 DISPLAY_BLACKBOARD

Aktualisiert den Inhalt eines Blackboards. Im gleichen Zuge wird die Aktualitätsinformation (Zeitstempel) aktualisiert.

3.2.1 Parameter

Name Name des Blackboards

Daten Die neue Information, die vorgehalten werden soll.

3.2.2 Rückgabe

Erfolg oder Misserfolg:

- Blackboard erfolgreich aktualisiert
- Blackboard existiert nicht
- Ungültige Parameter
- Weitere interne Fehler

3.3 CLEAR_BLACKBOARD

Löscht den Inhalt eines Blackboards. Blackboard ist nicht mehr Gültig.

3.3.1 Parameter

Name Name des Blackboards

3.3.2 Rückgabe

Erfolg oder Misserfolg:

- Blackboard erfolgreich aktualisiert
- Blackboard existiert nicht
- Ungültige Parameter
- Weitere interne Fehler

3.4 READ_BLACKBOARD

Ließt den Inhalt eines Blackboards aus. Zusätlich wird die Gültigkeit der Daten signalisiert. Wenn die Nachricht veraltet ist wird diese Information zurück gegeben.

3.4.1 Parameter

Name Name des Blackboards

3.4.2 Rückgabe

- Inhalt des Blackboards.
- Gültigkeitsinformation (Gültig/Ungültig)
- Erfolg oder Misserfolg:
  - Blackboard erfolgreich gelesen
  - Blackboard existiert nicht

  - Ungültige Parameter
  - Blackboard ist leer
  - Weitere interne Fehler

3.5 GET_BLACKBOARD_STATUS

Gibt den aktuellen Status eines Blackboards zurück.

3.5.1 Parameter

Name Name des Blackboards

3.5.2 Rückgabe

- Gefüllt/ Leer Indikator
- Gültigkeitsinformation (Zeitstempel der Aktualisierung)
- Gültigkeitsinformation (Gültig/ Ungültig)
- Erfolg oder Misserfolg:
  - Blackboard Status erfolgreich gelesen
  - Blackboard existiert nicht
  - Ungültige Parameter
  - Weitere interne Fehler

3.6 LIST_BLACKBOARDS

Listet alle vorhandend Blackboards auf.

3.6.1 Parameter

Keine.

3.6.2 Rückgabe

- Liste der erstellten Blackboards (Namen).
- Erfolg oder Misserfolg:
  - Blackboard Liste erfolgreich zurückgegeben
  - Weitere interne Fehler

3.7 DELETE_BLACKBOARD

Löscht ein Blackboard.

3.7.1 Parameter

Name Name des Blackboards

3.7.2 Rückgabe

Erfolg oder Misserfolg:

- Blackboard erfolgreich gelöscht
- Blackboard existiert nicht
- Ungültige Parameter
- Weitere interne Fehler

3.8 DELETE_ALL_BLACKBOARDS

Löscht alle Blackboards.

3.8.1 Parameter

Keine.

3.8.2 Rückgabe

Erfolg oder Misserfolg:

- Blackboards erfolgreich gelöscht
- Weitere interne Fehler

4 Erwartungshaltung

4.1 Erzeugte Programme (Client/ Server)

- Vollständigkeit der geforderten Funktionalität:
  - Umsetzung der geforderten Funktionen
  - Fehlerbehandlung
- Qualitative Anforderungen an das Verteilte System:
  - Verteilungstransparenz
    - Zugriffstransparenz
    - Nebenläufigkeitstransparenz
    - Fehlertransparenz
  - Offenheit (im Sinne Dokumentation der Schnittstellen)

- Weitere Nicht-Funktionale Anforderungen
  - Robustheit gegenüber externe Einflüsse
  - Qualität der Quelltextes
    - Verwendete Konstrukte in der Implementierung
    - Verständlichkeit der Implementierung
    - Dokumentation der Implementierung

4.2 Dokumentation

- Architekturbeschreibung
- Beschreibung der Verwendeten Externen Bibliotheken und deren Funktionalität
- Dokumentation der der Schnittstelle Server/ Client
- Dokumentation des Buildprozessess von Client/ Server
- Dokumentation der Verwendung von Client/ Server
- Dokumentation des Testumfanges und der -ergebnisse

5 Vorgehen

1. Teambildung (Gruppen a 6 Personen)
2. Vorstellung des Vorgehen am 18.05.2022 (Grobe Architektur, Absprache der Entwicklungsumgebung)
3. Umsetzung
4. Abgabe (19.06.2022 End-of-Day)