

Proiect CPD

Implementați un sistem care să simuleze o conversație între mai mulți oameni: o de dezbatere paralelă pe mai multe subiecte.

Regulile sunt

- trebuie să existe cel puțin 3 topicuri de dezbatere simultan
- fiecare persoană trebuie să participe la cel puțin două topicuri
- o singură persoană poate povesti la un moment dat
- o persoană poate povesti maxim 10 secunde o dată, apoi vine rândul următorului

Sistemul poate simula situația reală prin mai multe aplicații interconectate. Toate aplicațiile pot asculta topicurile de care sunt interesate. Singura aplicație care poate povesti e cea care deține un "token". Token-un trebuie să se plimbe printre aplicații după ce acestea termină de povestit, dar fără să povestească mai mult de 10 secunde fiecare.

Pentru implementare

- transmisia token-ului printre aplicații se poate face prin sockets
- puteți organiza aplicațiile într-un inel
- fiecare aplicație poate comunica cu cea din "dreapta" și "stânga" sa prin sockets.
- prin sockets vom transmite un token
- transmiterea mesajelor se va face prin pub/sub
- aplicația care are token-ul poate face publish
- aplicațiile care nu au token, nu pot face publish (pe perioada în care nu au token-ul), vor face doar subscribe
- vom avea $n \geq 3$ topics pe care se poate face publish (X, Y, Z, alfa, beta, gama....)
- fiecare aplicație trebuie să facă subscribe la cel puțin două subscriptions
- fiecare aplicație trebuie să facă publish pe cel puțin două topics
- fiecare aplicație va avea o "fereastră" (web/desktop/logguri) pentru fiecare subscription și/sau topic
- când o aplicație (cu token) face publish la un mesaj, vom vedea asta în "fereastra" pentru acel topic
- când o aplicație (fără token) primește un mesaj pe un anumit subscripțion, mesajul va fi vizibil în acea "fereastră"
- tokenul poate fi ținut maxim 10 secunde de o aplicație

Exemplu de implementare:

(socket) <- Aplicație -> (socket)

5050 <- Timotei -> 6060 <- Andreea -> 7070 <- Claudiu -> 5050

3 topics: Sport, Cooking, Harry Potter

Timotei face publish pe Sport și Cooking

Timotei face subscribe Cooking și Harry Potter

Andreea face publish pe Sport și Harry Potter

Andreea face subscribe pe Cooking, Sport și Harry Potter

Claudiu face publish pe Cooking, Sport și Harry Potter

Claudiu face subscribe pe Cooking și Harry Potter

--> Timotei primește (deține) token-ul de la Claudiu

- Timotei poate face publish pe Sport și Cooking.

- Andreea primește mesajele de pe Sport

- Andreea primește mesajele de pe Cooking

- Claudiu primește mesajele de pe Cooking

--> Timotei trimite token-ul mai departe la Andreea (fie că a terminat ce a avut de spus, fie că i-a expirat timpul)

--> Andreea deține token-ul (primit de la Timotei)

- Andreea face publish pe Sport

- Andreea face publish pe Harry Potter

- Timotei face primește mesajele de pe Harry Potter

- Claudiu primește mesajele de pe Harry Potter

"ferestrele"

--- PUBLISH ---

- mesaje luate din consolă (introduse de utilizator)

<<topic>>:

<<payload>> eg: Sport: Azi mergem la basket

- mesaje luate dintr-o fereastră de swing

o

fereastră per topic

- mesaje luate din ceva web page

un input per topic

--- SUBSCRIBE ---

- mesaje afișate în consolă

(preferabil logguri <- ca să apară și clasa)

<<clasa>>: <<topic>>: <<mesaj>>

- msaje afișate într-o fereastră de swing o fereastră per topic. publish se face în aceeași fereastră ca și subscribe dacă subscription-ul e pentru acel topic (Sport cu Sport, o fereastră)

- mesaje afișate în ceva pagină web

!!!

când aplicația nu are token-ul nu ar trebui să lase utilizatorul să introducă date sau să ignore datele introduse

1. Implementat sokets (verificat)

2. Pasat token

3. Limitat timp deținere token

4. Comunicare cu "fereastră" (când expiră cele 5-10-15 secunde, utilizatorul nu mai poate scrie în consolă sau nu sunt procesate mesajele, decât când revine token-ul)

5. Implementat publish

6. Legat publish cu token și consolă (să fie afișat ce trimiteți)

7. Implementat subscribe

8. Legați subscribe de consolă

9. Aveți grijă să nu vă citiți propriile mesaje (<- salvez local mesajul, când fac publish dacă e la fel cu ce am trimis chiar înainte, nu îl afișez)

10. Primiti un 10

