SimplePubSubTopicBroker&Client

Manoliu Victor-Codrin – grupa A7

1. Introducere

Se doreste realizarea implementarii unei aplicatii formata din trei componente : server, client de tip broker si client de tip publisher pentru a realiza un sistem de tip subscribe and publish. Clientul de tip publisher va publica un anumit articol caruia ii va atribui un topic predefenit iar clientul de tip subscriber va avea o lista de topicuri care le urmareste si va putea citi articolele corespondente topicurilor respective cu ajutorul server-ului ce va face acest match intre topicuri si articole.

2. Tehnologii utilizate

Ca sistem de compilare vom utiliza Makefile atat eficientei in rulare datorata de acesta cat si a comfortului din punct de vedere al factorului uman. Ca si protocol de comunicare se va folosi protocolul TCP/IP deoarece viteza de trimitere a datelor poate fi sacrificata intr-o masura considerabila in scopul sigurantei trimitelor datelor intre client si server, in cazul nostrum a articolelor si a posibilitatii primirii unor mesaje in cazul in care unele date nu s-au putut trimite.

2.1 Server

Serverul este unul concurrent implementat prin crearea unui nou procesil copil la fiecare noua conexiune si socket-uri de tip BSD vor fi utilizati.

Datele schimbate de user si client vor fi initial stocate in format char, mai precis sub forma unor array-uri de char , de exemplu : userName[MAXUSERLENGTH] unde MAXUSERLENGTH este un macro definit cu rolul de a specifica lungimea maxima pe care un utilizator o poate avea la user-ul sau iar apo acestea vor fi scrise intr-o baza de date ce foloseste MySQL si hostata pe domeniul facultatii (fenrir.info.uaic.ro).

2.2 Client

Pentru a realiza conexiunea cu server-ul vor fi folositi socket-uri de tip BSD. Toate informatiile trimise si primit de catre server for fi insotite de un mesaj de confirmare in caz ca informatia a

fost trimisa si primita cu success. Informatiile respective cat si mesajele de confirmare/infirmare vor fi afisate in consola, client-ul nedispunand de o interfata grafica.

3. Arhitectura aplicatiei

3.1. Participanti ai aplicatiei

Serverul.

Va face listen la portul 2025 pentru conexiuni din partea clientilor. Cand acesta primeste o conexiune, un nou process copil va fi creat cu ajutorul primitei fork() iar valoarea returnata de aceasta va fi interogata corespunzator. Imediat ce un client s-a conectat la server acesta va fi rugat sa se autentifice pentru a se putea stabili rolul acestuia si implicit ce drepturi are. Va primi de la client user-ul si parola cu care acesta a fost inregistrat si va verifica aceste credentiale prin intermediul tabelei **user** prezenta in baza de date mentionata ce va contine ca si coloane atat user-ul parola cat si rolul fiecarui user(1 = publisher, 2=subscriber). Daca autentificarea s-a realizat cu success, in functie de rolul clientului, serverul va trimite la client un token de forma **logat_publisher** sau **logat_subscriber** iar pe baza acestui token se va decide daca user-ul chiar este logat si comenzile la care acesta are acces.

Va primi de la clientii publisher: topicul articolului pentru care acestia vor dori sa posteze , titlul articolului si textul acestuia, va stoca initial fiecare dintre aceste informatii intr-un array , fiecare informatie fiind separate de un separator, va sparge aceste informatii stocandu-le in cate un array de tip char iar apoi le va scrie in tabela **articole.**

Va primi de clientii subscriber: topicul articolelor pe care acestia doresc sa le vizualizeze si va cauta in baza de date dupa topicul respective afisand titlul articolelor si textul acestora.

Clientul

Va comunica cu server-ul la portul 2025

Client-ul publisher va trimite topicul articolului, titlul si textul acestuia prin comanda **publish**

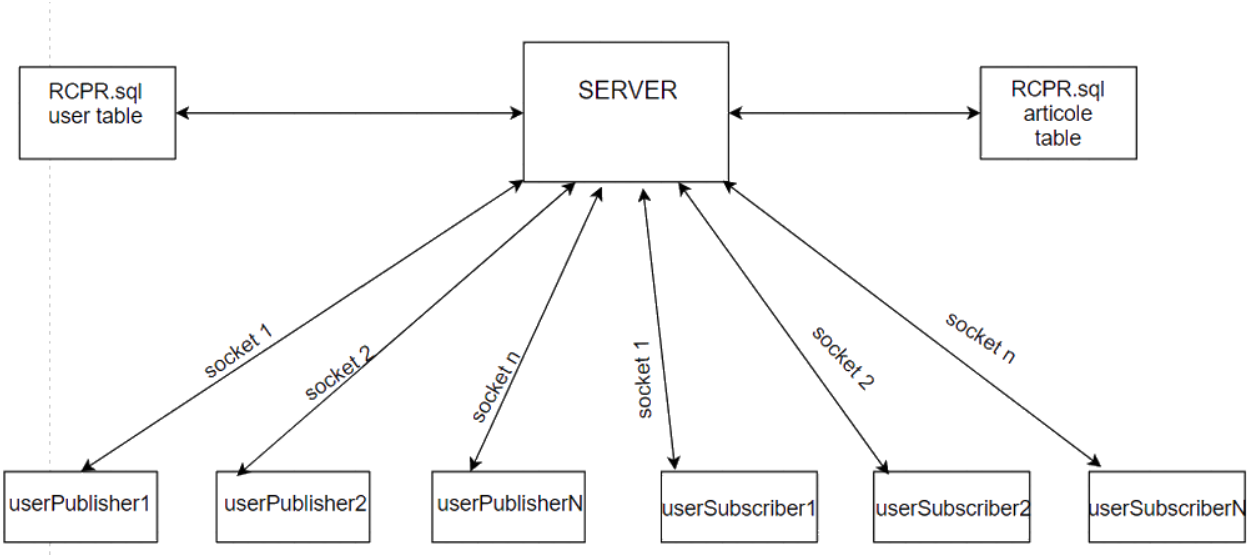
Client-ul subscriber poate lista toate articolele ce apartin unui anumit topic prin apelarea comenzii din protocol: **subscribe**

Logarea se va realiza prin comanda : **login**

Logout-ul se va realiza prin comanda : **logout**

Deconectarea unui client in cazul in care acesta doreste acest lucru se realizeaza prin executarea comenzii : **quit** iar server-ul va fi notificat de inchiderea client-ului printr-un mesaj.

3.2 Diagrama detaliata a aplicatiei



3.3 Protocol disponibil

register : va permite clientului sa inregistreze . Va fi cerut user-ul, parola si rolul care acesta doreste sa il aiba prin intermediul apelului blocant de read si apoi acestea vor fi scrise la server prin intermediul apelului blocant de write care apoi vor fi scrise in baza de date mentionata.

login : va cere clientului sa introduca numele de utilizator si parola cu care este inregistrat. Credentialele sale vor fi cautate in baza de date, in tabela useri.

publish : va cere clientului (daca acesta este de tip Publisher) sa introduca topic-ul articolului pe care doreste s ail posteze iar apoi textul efectiv al acestuia

subscribe : va cere clientului (daca acesta este de tip Subscriber) sa specific topic-ul din care acesta doreste sa vizualizeze articolele

logout : va permite clientului indifferent de rolul acestuia sa se delogheze si sa se poata reloga din nou cu un alt user si parola ce au alt rol sau chiar sa se inregistreze cu un nou username si parola si un rol la alegerea sa

quit: va inchide conexiunea cu server-ul pentru client-ul current

Datele trimise de la client la server vor fi stocate in baza de date mentionata la sectiunile anterioare. A fost aleasa o baza de date datorita eficientiei lucrarii cu aceasta, posibilitatea stocarii a mai multor informatii ce permit interogarea si rezolvarea bug-urilor sau erorilor, pentru securitatea datelor (baza de date fiind pe un domeniu securizat) si pentru faptul ca articolele nu vor fi pierdute dupa inchiderea serverului fie ca este o inchidere dorita sau fortata.

4. Concluzii

Aplicatia poate fi imbunatatita prin realizarea unui sistem de comunicare concurrent ce va folosi primitivei select in scopul eliminarii riscului unui comportament nedeterminist din partea server-ului la realizarea conexiunilor cu clientii.

Ar putea fi implementata o interfata grafica pentru a usura vizualizarea si postarea de noi articole.

Bibliografie

Documentatie Socket BSD :https://profs.info.uaic.ro/~computernetworks/cursullaboratorul.php

Documentatie Make : http://www.cs.colby.edu/maxwell/courses/tutorials/maketutor/

Documentatie MySQL C API: https://dev.mysql.com/doc/refman/5.7/en/c-api.html