RPC-BASED CHAT

Pràctica de projectes en arquitectura distribuïda

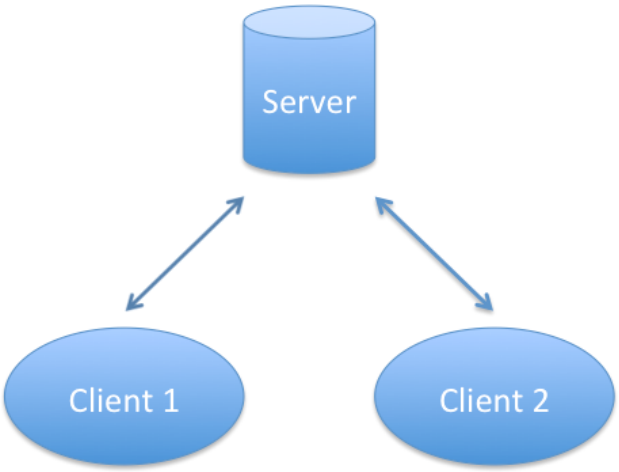
Curs 2015-2016

Balbina Virgili Rocosa
ls27476

Índex

1-. Introducció

L'objectiu principal d'aquesta pràctica és implementar una aplicació distribuïda client-servidor que implementi un xat, a partir de la utilització de Remote Procedure Calls.



El servidor ha d'oferir al client els mètodes necessaris per llegir i escriure nous missatges en el chat i, a més a més, ha de mantenir un fitxer local amb aquests missatges.

Per altra banda, el client ha de poder crear nous missatges que no podrà mantenir en cap fitxer local i, per tant, haurà d'anar demanant aquesta nova informació al servidor, mitjançant els mètodes dels que aquest té disponible.

2-. Disseny general del sistema

Per tal de poder implementar aquesta aplicació distribuïda s'ha utilitzat l'eina rpcgen, ja que permet implementar les crides que fan els clients als mètodes del servidor d'una manera senzilla.

Així doncs, el primer que s'ha realitzat ha sigut crear un fitxer anomenat xat.x amb la següent informació:

```
struct Phrase {                          //Estructura creada per retornar
        char content[300];                la informació llegida del servidor
};

program xatClient {                                  //Nom del programa
        version primeraVersio {                      //Nom de la versió
                int write(Phrase phrase) = 1;        //Mètodes a oferir
                Phrase getChat() = 2;
        } = 1;                                       //Número de la versió
} = 0x20000000;                                      //Referència del Servidor
```

Totes les funcions definides només admeten un únic paràmetre, el qual pot ser una estructura o un tipus simple. És per això que s'ha creat l'estructura Phrase, tant per enviar la nova informació com per rebre-la.

Una vegada definit aquest fitxer, s'ha executat la comanda següent:

rpcgen –a xat.x

D'aquesta manera s'han creat els 7 fitxers necessaris per la connexió que són els següents:

Makefile.xat, xat_xdr.c, xat.h, xat_server.c, xat_svc.c, xat_client.c, xat_clnt.c

A continuació, ha calgut implementar les funcions creades.

En el fitxer xat_client.c s'ha creat un nou fil d'execució a part del principal.
El principal dóna la benvinguda al client i escolta els diferents missatges que va introduint enviant-los al servidor, fins que aquest decideix finalitzar el xat i introdueix la paraula 'exit'.
Per tal d'enviar la nova informació al servidor utilitza la funció següent:

```
void write(Phrase phrase);
```

Per altra banda, el nou thread creat demana nova informació al servidor cada 10 segons, utilitzant la funció següent:

```
Phrase getChat();
```

El temps de 10 segons està implementat a partir d'un signal d'alarma que executa la seva rutina quan el temps especificat ha passat. En la seva rutina el que fa es desbloquejar un mutex que es queda bloquejat en la entrada de la lectura d'informació del servidor.
La funció getChar() retorna sempre tota la informació que té emmagatzemada en el fitxer i és el propi client que a partir d'una variable entera sap quina és la última informació que ja ha mostrat, i per tant, només torna a mostrar aquella que es nova per ell.

En el fitxer xat_server.c també s'han desenvolupat les dues funcions d'escriptura i lectura.
La funció d'escriptura el que fa es rebre el nou contingut del client i escriure'l dins del fitxer local.
En canvi, la funció de lectura el que fa és llegir tot el contingut del fitxer i retornar-lo al client.

Per últim, ha calgut compilar l'aplicació desenvolupada a partir de la comanda següent:

Make –f Makefile.xat

I perquè la compilació amb els threads fos satisfactòria ha calgut afegir la següent línia al fitxer Makefile.xat:

LDLIBS += -lpthread

3-. Proves realitzades

Per tal de comprovar el funcionament de l'aplicació implementada, s'ha obert la aplicació amb tres shells diferents.

- La primera executa el servidor:

  xat_server
- La segona executa el client amb nickname client1:

  xat_client localhost client1
- La tercera executa el client amb nickname client2:

  xat_client localhost client2

A partir d'aquesta execució s'ha comprovat que la informació que introdueix qualsevol dels 2 clients arriba al server i es també enviat a l'altra client, quan aquest fa la petició. Una vegada acabada l'execució, tota la conversa s'ha emmagatzemat al fitxer local del servidor (xat.txt).

4-. Problemes observats i conclusions

La utilització de l'eina rpcgen ha ajudat molt al desenvolupament d'aquesta aplicació distribuïda, ja que no ha calgut preocupar-se de la configuració de les connexions paral·leles amb el servidor i només s'ha hagut de desenvolupar les funcionalitats desitjades en els dos casos.

Tot i això, s'han trobat problemes a l'hora de definir les funcions, ja que s'intentava enviar i retornar un char * pel contingut llegit i escrit. I, per tant, el servidor només rebia un char i no tota la cadena de caràcters que se li enviava. El mateix passava al inrevés, fins que s'ha substituït aquesta variable no simple per una estructura, i el contingut de l'estructura se li ha fixat un valor exacta, ja que sinó tampoc arribava correctament la informació.

També hi ha hagut problemes a l'hora de compilar els threads, ja que donava un error que no reconeixia les seves funcions. És per això que s'ha introduït la línia (*LDLIBS* += *-lpthread*) en el fitxer Makefile.xat.

Una vegada finalitzada la pràctica es pot dir que aquesta eina serveix de gran ajuda pel tipus d'aplicacions distribuïdes client-servidor, tot i que crec que s'ha de conèixer bé la eina per tal d'utilitzar-la, ja que els problemes que s'han trobat al desenvolupar el chat són propis del seu ús i no de la programació en c en si.