

Memoria Bloc 3 AST

Daniel Vilardell

Índex

1	Practica 7	2
1.1	Resum general establiment connexió	2
1.2	Resum general alliberament connexió	2
1.3	Protocol	3
1.4	TSocket	4

Aquest document no sera molt detallat ja que el codi en si es prou entenedor i esta comentat lo suficient com per veure el que fa cada secció.

1 Practica 7

1.1 Resum general establiment connexió

Primer explicarem una mica com funciona l'establiment de connexió entre el servidor i el client i després explicarem els diferents metodes de les classes TSocket i Protocol.

En primer lloc dins del main es crea un Host servidor. Aquest crida el metode openListen i crea un socket passiu que tindrà la funció de crear les connexions entre els clients i el servidor. Un cop fet això el servidor entra en un bucle de while true on crida la funció accept que el mantindrà adormit fins a que no rebí una petició de connexió per part de un client.

Per la banda del client, un cop ja s'ha obert el servidor crida la funció openConnect passant el port del servidor com a parametre, i aquesta funció creara un nou socket encarregat de gestionar aquesta connexió i cridara la funció connect. Aquesta ja començara el tema de enviar i rebre paquets per tal d'establir la connexió. En primer lloc introdueix el socket dins de la llista dels sockets actius ja que estara en connexió directa durant la fase de intercanviar dades. Despres envia un segment amb el flag de syn activat per a establir la conexio i canvia el estat a SYN_SENT. Un cop fet això s'adorm fins a rebre la resposta.

Aquest segment de syn enviat arriba al socket de listen del servidor. Alla es crea un nou socket per a posarlo als sockets actius i ell s'encarregara de la connexio amb el client, així el socket de listen pot seguir executant la seva funció. Es posa també el segment a la cua de accepts i es desperta el fil dormit que esperava la connexió. Aquest fil retornarà aquest tsocket així el host pot guardarlo i mes tard tenir la opció de tancar aquesta conexió. Aquest thread es posa al estat de conexió establerta i envia un segment amb el flag de syn com a resposta.

Quan el segment de syn torna a arribar al client aquest actualitza el estat a connexió establerta i desperta el fil dormit que esperava la resposta.

Despres d'això ja tenim la connexió entre el servidor i el client establerta. A partir d'aquí poden intercanviar segments sense problemes.

1.2 Resum general alliberament connexió

Per a alliberar la connexió entre un servei i el seu client hi ha dos formes diferents, que el servidor cridi el metode close o que ho faci el client. Les dos maneres son simetriques, el que s'intercanviarien els papers el servidor i el client. Per tant explicarem el cas en que el client demana de tancar la connexió i el altre es analog.

El client crida primer la funció close del socket que enviarà un segment amb el flag de fin activat al servidor i s'adormirà fins a rebre el fin de resposta. Aquest fin arribarà al servidor i aquest canviarà el estat a close wait i cridara la funció close. En aquest cas simplement crearà el segment amb el flag de fin activat, l'enviarà i tancarà la conexió definitivament eliminant el socket de la llista de sockets actius de Protocol.

Un cop aquest ultim segment arriba, el client despertarà el fil que esperava resposta i aquest tancarà també la conexió de la mateixa forma, es a dir, eliminant el socket de la llista de sockets actius de Protocol.

1.3 Protocol

- **openListen:** Crea un socket passiu encarregat de escoltar les peticions de establiment de connexió i de la creació de sockets actius.
- **openConnect:** Crea un socket per al client i comença el establiment de connexió.
- **addListenTSocket:** Afegeix a la llista un socket passiu.
- **addActiveTSocket:** Afegeix a la llista un socket actiu.
- **removeListenTSocket:** Elimina de la llista un socket passiu.
- **removeActiveTSocket:** Elimina de la llista un socket actiu.
- **getMatchingTSocket:** Retorna un socket actiu amb el port origen i destí especificats. En el cas que no existeixi retorna el socket passiu amb el port origen especificat.
- **portInUse:** Mira si el port passat com a parametra esta en us.
- **newport:** Retorna un port lliure.

1.4 TSocket

- **listen:** Inicialitza el socket passiu encarregat de gestionar les peticions de connexió al servidor.
- **accept:** S'espera fins que hi ha una petició de connexió i quan hi es retorna el socket encarregat de gestionarla.
- **connect:** Afegeix el socket als sockets actius del client i envia el segment de SYN. S'adorm fins a rebre la resposta.
- **close:** Si ets el que ha creat la petició de tancar la connexió envia un segment de fin i espera la resposta. Un cop rep la resposta elimina el socket de la llista de sockets. Si no ets el que ha creat la petició simplement respons amb un fin i elimines el socket.
- **processReceivedSegment:**
 - LISTEN: Si rep un segment de syn crea el socket actiu a la banda del servidor, l'afegeix a la llista de sockets actius i de cua d'acceptats i envia un segment de syn al client.
 - SYN_SENT: Si rep la resposta del segment de syn desprerta el fil que la estava esperant.
 - ESTABLISHED: Si rep un segment de fin crida close per a tancar la connexió.
 - FIN_WAIT: Desperta el fil que estava esperant la resposta del fin.
 - CLOSE_WAIT: En el diagrama un cop rep el fin envia directament el fin de resposta i tanca la connexió així que mai ens trobarem en aquest estat al rebre un missatge.
- **sendSegment:** Envia un segment pel canal.
- **stateToString:** Retorna un string amb el estat actual.