



Pràctica 1, Programació d'aplicacions de xarxa

Per: Jordi Arenas Romero
Escola Politècnica Superior

Treball presentat a: Enrique Guitart

Baraut

Xarxes

2022 - 2023

16 -04 -2023

Índex

Introducció	3
Estructura del client	3
Estructura del servidor	3
Implementació del client.....	4
Fase de registre	4
Fase de manteniment de comunicació	4
Fase d'enviament i recepció d'arxius.....	4
Implementació del servidor.....	4
Fase de registre	5
Fase de manteniment de comunicació	5
Fase d'enviament i recepció d'arxius.....	5

Introducció

En aquesta pràctica s'ha realitzat la implementació d'una comunicació entre el servidor i els clients utilitzant el llenguatge Python per a la implementació del servidor i C per a la implementació del client.

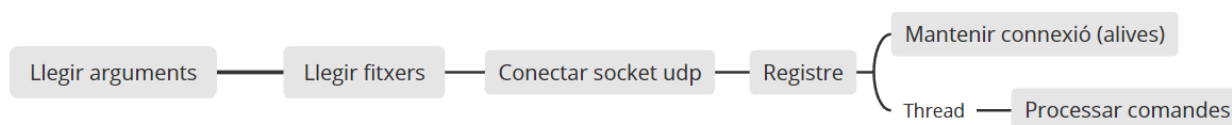
Tant la implementació del client com la del servidor estan dividides en quatre parts: La fase de registre, la fase de manteniment de la comunicació, la fase d'enviament d'arxius i la fase de recepció d'arxius.

En aquest informe s'exposaran les estratègies implementades tant pel servidor com pel client.

Estructura del client

Per a realitzar el client, s'ha emprat l'estructura que es pot veure en l'esquema 1. Cal remarcar la importància de crear un fil per a processar les comandes ja que, si ho féssim sense crear cap fil ni cap procés alternatiu es podrien ocasionar problemes en l'enviament dels *alives* si es tardés a processar alguna comanda.

Tanmateix, no és necessari crear un altre fil per cada comanda rebuda ja que cada client només pot executar una comanda al mateix temps.



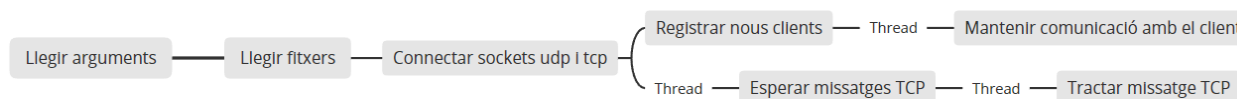
Esquema 1

Estructura del servidor

Per a realitzar el servidor, s'ha emprat l'estructura que es pot veure en l'esquema 2. En aquest cas, trobem el primer fil després de fer la connexió dels sockets, amb la intenció de poder, per una banda, seguir registrant als nous clients (tasca que ha d'estar supervisant constantment) i, per altre banda, esperarà la recepció de paquets TCP.

Cada vegada que un client es registri correctament, es crearà un nou fil que s'encarregarà de l'enviament i recepció dels alives amb aquest client. D'aquesta forma, el servidor sempre podrà atendre noves peticions de registre provinents de qualsevol client.

D'una forma anàloga, cada vegada que es rebí una instrucció per un socket TCP (ja sigui un SEND-FILE o un GET-FILE), es crearà un nou fil que s'encarregarà de processar el paquet rebut, incloent la correcta identificació de quin client l'ha enviat i l'enviament dels paquets necessaris per a respondre al client.



Esquema 2

Implementació del client

Per a gestionar el funcionament general del client s'ha utilitzat el diagrama d'estats que es pot veure en la Figura 1.

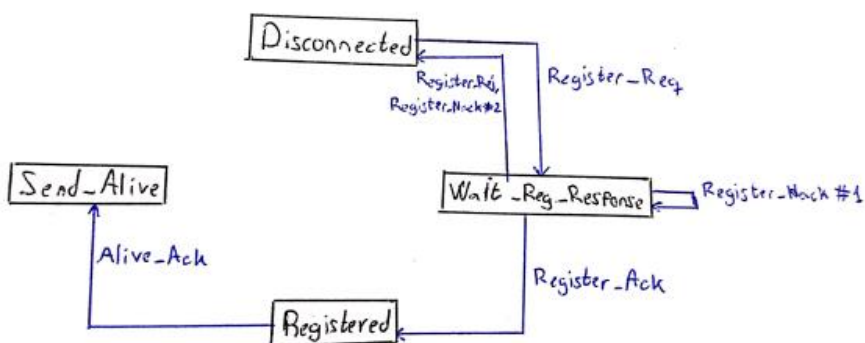


Figura 1

Fase de registre

Per a fer el registre s'han utilitzat dos bucles imbricats. El primer conta quants processos de registre s'han fet i el segon conta quants paquets s'han enviat. D'aquesta forma, quan es rep un [REGISTER_REJ] només s'ha d'assignar el valor màxim de nombre de registres i de paquets enviats. Quan es rep un [REGISTER_NACK] s'assigna el valor màxim de paquets enviats per a que es passi al següent procés de registre si no s'ha arribat al màxim i quan es rep un [REGISTER_ACK], a part d'assignar el valor màxim al nombre de registres i de paquets enviats es canvia l'estat del client a REGISTERED.

Fase de manteniment de comunicació

Per a la fase de manteniment de comunicació es fa un bucle que es manté actiu fins que finalitzi la comunicació amb el servidor.

Fase d'enviament i recepció d'arxius

Aquestes dues fases s'han realitzat de forma molt similar. Inicialment, per a llegir les comandes s'ha creat un fil que les executa sense alterar la fase de manteniment de comunicació. Seguidament, s'envia el paquet al servidor i es tracta la seva resposta. Si la resposta és correcta, s'envia o es rep la informació segons la comanda emprada.

Implementació del servidor

Per a gestionar el funcionament general del servidor s'ha utilitzat el diagrama d'estats que es pot veure en la Figura 2.

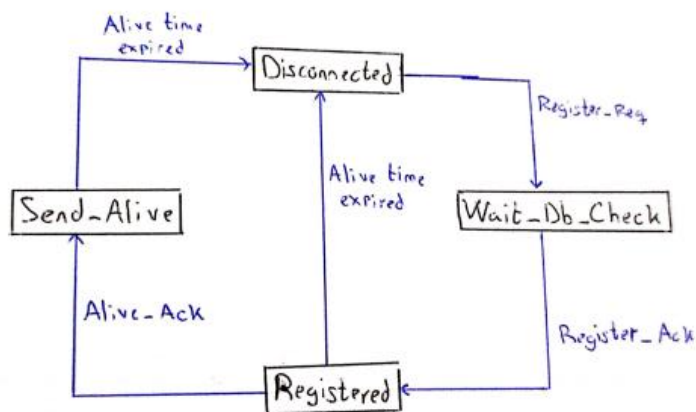


Figura 2

Fase de registre

Per a fer el registre del servidor s'esperen paquets UDP constantment i, en cas de rebre un, es crea un nou fil que el tractarà i comprovarà que totes les dades siguin correctes per, en cas afirmatiu, registrar el nou client com un objecte Client. Quan es fa el registre del client, es crea un nou fil que comprovarà que es rebi algun paquet [ALIVE_INF] en el temps preestablert.

Cal tenir en compte que els clients acceptats tenen un objecte Client ja creat (es crea quan es llegeix el fitxer d'equips acceptats) i, per tant, només caldrà modificar les seves dades.

Fase de manteniment de comunicació

El manteniment de la comunicació s'estructura de forma similar a la fase de registre en quant a la recepció del paquet. Una vegada s'identifica que es tracta d'un paquet [ALIVE_INF], es comprova que les dades siguin correctes i es crea un fil que constantment comprovarà que es rebin paquets [ALIVE_INF] en la freqüència preestablerta.

Fase d'enviament i recepció d'arxius

Novament, aquestes fases són molt semblants, doncs la única diferència és que en una es llegeix un fitxer i s'envien les dades mentre que en l'altre s'escriu un fitxer i es reben les dades. Per a que es puguin llegir els paquets TCP de forma constant, es crea un fil que crea un socket TCP i, quan es rebi un paquet de un dels clients assignats, es crearà un nou fil (per a poder seguir mantenint la comunicació via TCP amb els altres clients) que li assignarà aquest socket al client que l'ha enviat, per a poder controlar si ja està enviant o rebent l'arxiu de configuració.