XARXES

Pràctica 1 ‖ **Model client-servidor**

Curs 2018-2019

Edgar Moreno Molina

Índex

Índex de figures

1 Estructura client i servidor

Cal tenir en compte les següents consideracions:

- Hi apareixen:
  - Funcions representades en color negre i emmarcades.
  - Estats del client en color blau.
  - Operacions XOR i AND en color vermell.
- Dins de cada funció es mostren el possibles canvis d'estat del client, fet que no representa un canvi d'estat obligatori, pot ser que aquest no canviï d'estat i per tant, segueixi en el mateix estat anterior.
- Les funcions iniciades en el mateix instant de temps queden representades mitjançant branques adjacents i l'operació AND.
- El símbol *EXIT* en majúscules i color verd simbolitza la finalització de l'execució en el client o servidor.

1.1 Diagrama blocs client

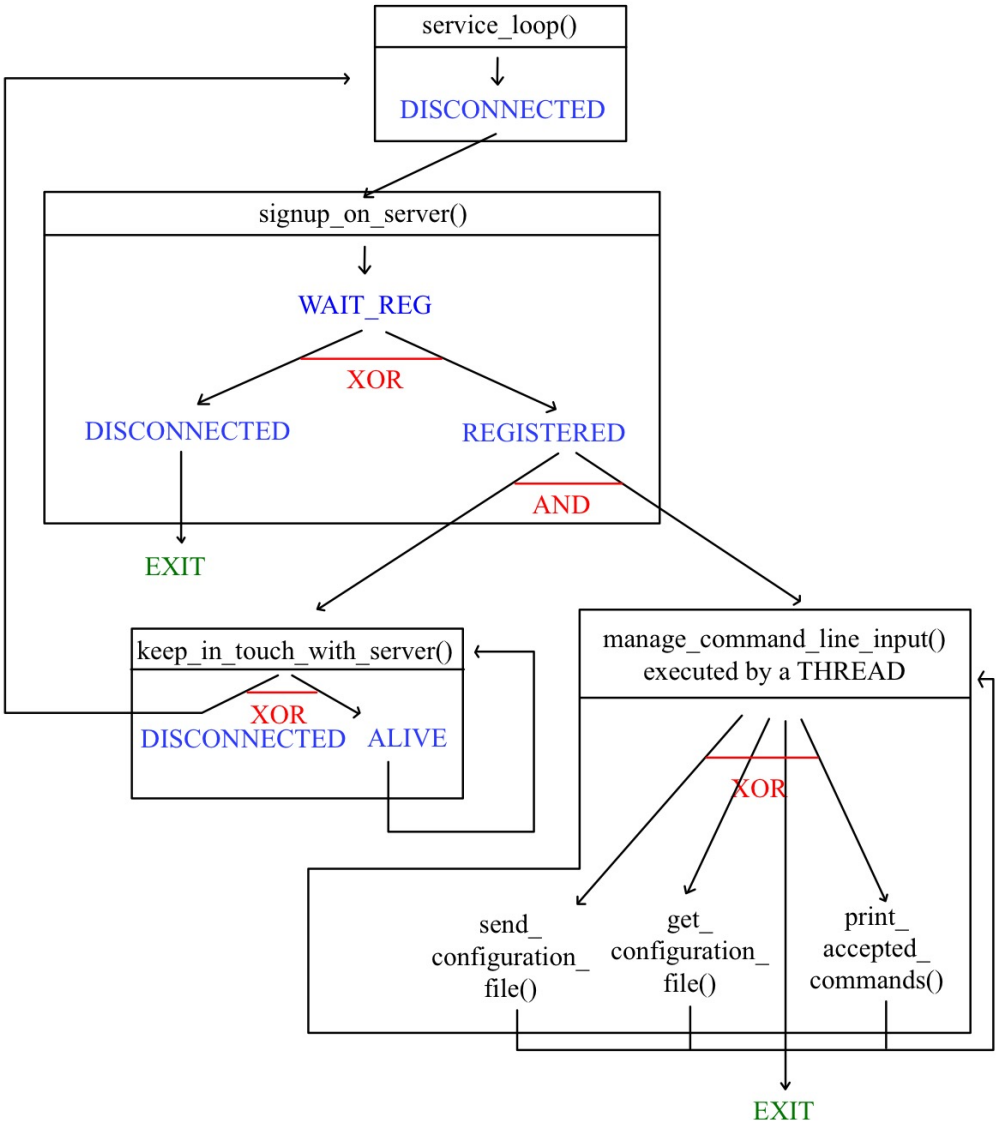

Figura 1: Diagrama blocs client

1.2 Diagrama blocs servidor

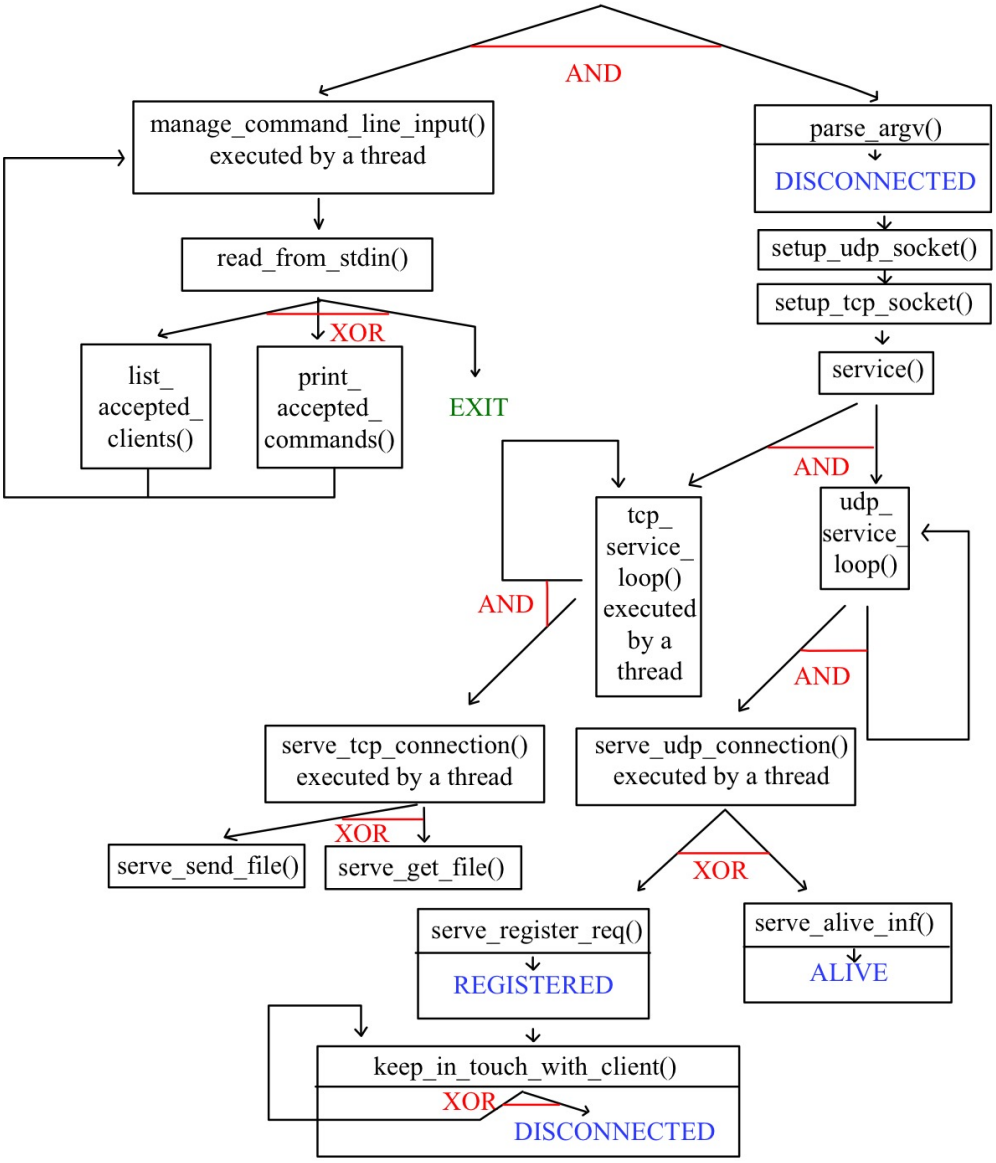

Figura 2: Diagrama blocs servidor

2 Estratègia emprada manteniment comunicació

2.1 Client

El manteniment de la comunicació en el client, una vegada el client ha estat registrat en el servidor, s'ha gestionat mitjançant una única funció ***keep_in_touch_with_server()*** que **s'encarrega** d'**enviar i esperar** durant un cert **timeout** la **resposta del servidor**.

Per a la **rebuda** de **paquets** resposta als paquets enviats s'ha utilitzat la funció ***select()*** que permet **monitorejar** el descriptor de fitxer o **socket durant** un cert **timeout màxim**.

Per tal d'**enviar** un **paquet ALIVE_INF cada r segons** s'ha **tingut en compte** el **temps esperat** en la **rebuda** de la **resposta a l'anterior ALIVE_INF enviat**, **mitjançant** els atributs ***tv_sec*** i ***tv_usec*** de l'estructura ***timeval***, resultants de l'operació *select* en el socket. (Evidentment, en el primer paquet enviat no s'ha tingut en compte el temps esperat en la rebuda de la resposta a l'anterior paquet enviat, doncs aquest anterior paquet enviat, no existeix).

2.2 Servidor

El manteniment de la comunicació en el servidor, una vegada rebut un paquet REG_REQUEST correcte, s'ha gestionat mitjançant **2 *daemon threads*** específics **per** a **cada client**, que s'executaran simultàniament:

- El primer s'encarregarà de **processar i respondre a** tots els **paquets ALIVE_INF rebuts** executant la funció *serve_alive_inf()* i informarà al segon thread (detallat a continuació) de la rebuda d'un paquet ALIVE_INF del client corresponent (mitjançant l'assignació de la variable client.is_alive_received a *True*).
- El segon s'encarregarà de **gestionar** els **temps** de **rebuda** de **paquets ALIVE_INF** per al client corresponent accedint al valor de la variable *client.is_alive_received*, la qual serà modificada pel thread anterior en rebre un paquet ALIVE_INF. El còmput serà dut a terme en la funció *keep_in_touch_with_client()*.

3 Diagrama estats protocol implementat sobre UDP

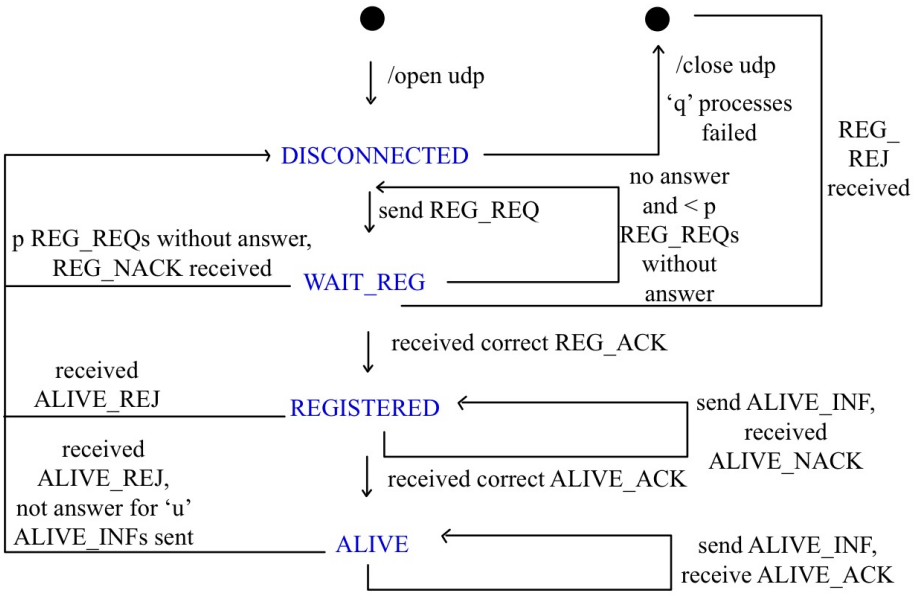

Figura 3: Diagrama estats protocol implementat sobre UDP

4 Consideracions i decisions preses

4.1 Indeterminisme

En el cas que venci un temporitzador sobre un paquet i es rebi el paquet en un instant molt pròxim de temps, el servidor determinarà amb exactitud quin esdeveniment del dos s'ha produït abans. Però en cas que es **venci el temporitzador** sobre un **paquet** i es **rebi** el **paquet** en el **mateix instant** de temps, es produirà una situació indeterminista. Per exemple, partint del test número 2 del servidor (especificat en el client):

```
-t 2: No s'envien [ALIVE_INF] corresponents a les seqüències: 3, 4, 6, 8
, 9, 10 i 11.
```

Es pot donar el cas que el temporitzador del 5è paquet i la rebuda del 5è paquet siguin dos esdeveniments que es produeixen en el mateix instant de temps. En aquesta situació el comportament del servidor pot ser qualsevol dels detallats a continuació:

- El servidor considera que el 5è paquet ha arribat abans de finalitzar el temporitzador, per tant aquest s'acceptarà i el client passarà a l'estat ALIVE .
- El servidor considera que el temporitzador acaba abans de rebre el paquet i per tant com els paquets 3 i 4 no han sigut rebuts (ni enviats), el servidor determina que s'han perdut 3 paquets ALIVE_INF consecutius i per tant el client passa a l'estat DISCONNECTED.

El servidor proporcionat pels professors respon a aquest criteri i en la implementació també he considerat aquest criteri com el més correcte, no tan sols s'ha seguit en la part del servidor sinó també en la part del client.

4.2 Respostes a paquets incorrectes

En cas que el **servidor rebi** un **paquet erroni**, aquest **respon** amb un **paquet** on tots els **camps** menys el camp data són **zeros** i el camp data indica el motiu de rebuig. S'ha continuat amb aquesta tendència d'**evitar compartir informació innecesaria en** el **camp data** i els motius de rebuig són molt generals, per exemple: *Wrong data* (dades incorrectes) i no del tipus *Incorrect random number* (nombre aleatori incorrecte). Aquesta decisió ha sigut presa per motius de seguretat.

4.3 Consideracions generals

- S'han considerat 3 tipus de outpus:
  - INFO: Missatge informatiu (per exemple, mode debug iniciat).
  - DEBUG: Missatge informatiu que es mostrarà quan el mode debug està activat (per exemple, llegits paràmetres d'arxiu de configuració).
  - ERROR: Errors de funcionament (per exemple, obertura incorrecta de fitxers).

  Els missatges INFO i ERROR és mostraran independentment de si el mode debug és activat o no.
- No importa l'ordre dels arguments introduïts en la línea de comandes: és el mateix *./client -d -c client1.cfg* que *./client -c client1.cfg -d*.
- Si no es poden obrir els fitxers proporcionats en la línea de comandes, es mostrara un missatge d'error i s'intentaran obrir els fitxers per defecte. Si els fitxers per defecte no poden ser oberts es mostrarà un missatge d'error i es finalitzarà l'execució.
- S'ha considerat que els arxius proporcionats per l'execució tindran la sintaxisą correcta i seguiran amb l'estructura descrita a la documentació.
- En cas d'introduir una comanda incorrecta en el client o servidor, es mostrarà el missatge d'error corresponent i a més és mostraran les comandes disponibles (list i quit en el servidor i get-conf, send-conf i quit en el client) i una concisa descripció d'aquestes.