

Activitat No Presencial

Programació de Sockets

Volem portar el registre d'alumnes que assisteixen al laboratori de XC. Per tal de dur a terme aquesta tasca, muntem un particular servidor amb el qual els alumnes es poden comunicar, i que funciona de la següent manera:

- El servidor rep un datagrama UDP en el port 8080, enviat des d'una màquina client X (la de l'alumne). En aquest datagrama, el client envia una tira de caràcters que representa un nombre enter (p.ex 9090)
- El servidor, a través d'un nou socket, estableix una connexió TCP amb la màquina client X en el port indicat en el contingut del datagrama UDP
- A través de la connexió TCP establerta l'equip del client envia el nom d'un alumne al servidor. Quan el servidor rep el nom de l'alumne el mostra per la pantalla.

Es demana que escriviu el codi **en C** d'un client capaç d'interactuar amb el servidor descrit anteriorment. El client s'invoca amb els següents paràmetres de línia de comandes:

```
$ client <NOM_MÀQUINA_SERVIDOR>
```

El codi del servidor es proporciona en forma d'un fitxer jar que es pot executar de la següent manera:

```
$ java -jar servidor.jar
```

Capturar els missatges intercanviats entre el client i el servidor amb tcpdump. Si ambdós s'executen en una mateixa màquina, hi ha que capturar els paquets en la interfície loopback (lo):

tcpdump -vs 1500 -ni lo

Executa la següent comanda immediatament després de comunicar el client i el servidor. Identifica el socket que s'ha creat com a conseqüència de la connexió, i que està en TIME_WAIT.

netstat -nat

Què entregar?

No hi ha que entregar res.

El treball s'avaluarà amb unes preguntes tipus test relacionades amb la pràctica que es faran el dia de l'examen final.

A continuació hi ha un exemple del tipus de preguntes que es faran per l'avaluació de la pràctica.

Exemple del tipus de preguntes que es demanaran en l'avaluació de la pràctica:

Disposem d'una sistema (amb una sola interfície de xarxa, amb adreça 150.12.27.42) executant el següent codi:

```
#include <sys/socket.h>
#include <netinet/in.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MIDABUFFER 256

1 int main(int argc, char *argv[])
2 {
3     char    buf[MIDABUFFER];
4     int     sdfS, sdfR, addrlen;
5     struct  sockadr_in sinS, sin2;
6     int     llegit;

8     memset(&sinS,0,sizeof(sinS)); memset(&sin2,0,sizeof(sin2));
9     if((sdfS = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
10         perror("socket");
11         exit(1);
12     }
13     sinS.sin_family = AF_INET;
14     sinS.sin_addr.s_addr = inet_addr("127.0.0.1");
15     sinS.sin_port = htons(8080);
16     if(bind(sdfS, (struct sockaddr *)&sinS, sizeof(sinS)) == -1) {
17         perror("bind"); exit(1);
18     }
19     if(listen(sdfS, 10) == -1) {
20         perror("listen"); exit(1);
21     }
22     addrlen = sizeof(sin2);
23     if((sdfR = accept(sdfS, (struct sockaddr *)&sin2, &addrlen)) == -1) {
24         perror("accept"); exit(1);
25     }
26     while(1) {
27         if((llegit = read(sdfR, buf, MIDABUFFER)) == -1) {
28             perror("send"); exit(1);
29         }
30         if(llegit > 0) {
31             printf("Rx (%d): %s", llegit, buf);
32         } else {
33             break;
34         }
35     }
36     close(sdfR) ; close(sdfS) ;
37 }
```

Digues quines de les següents respostes són certes respecte el codi del servidor anterior:

- 1) Fa servir sockets TCP (cert)
- 2) Quan s'executa el programa es quedarà bloquejat en la línia 23 fins que es connecti un client (cert)
- 3) És possible obtenir següent bolcat després d'haver executat el programa (cert)

```
> netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.1:8080          0.0.0.0:*               LISTEN
```

- 4) És possible obtenir següent bolcat després d'haver-se connectat un client (cert)

```
> netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.1:8080          0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:8080          127.0.0.1:55848        ESTABLISHED
```

- 5) En el client s'haurà de fer la crida al sistema connect per establir la connexió amb el servidor. (cert)
- 6) Si s'executa dos cops el servidor s'obtindrà el següent bolcat: (fals)

```
> netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.1:8080          0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:8080          0.0.0.0:*               LISTEN
```