

# Explicació SOCKETS

## Conceptes:

- ports
- Socket de Berkeley o Internet socket.

## System calls:

Ajuda: **man <system\_call>**

p.e.: **man socket**

### ● **int socket(int domain, int type, int protocol)**

Crea un socket. Aquesta crida ens retorna el descriptor que identificarà al socket.

- **domain:** Família de protocols (p.e. **PF\_INET**: TCP/IP, **PF\_UNIX**: Unix).
- **type:** Tipus de comunicació:
  - **SOCK\_STREAM** Servei orientat a connexió per flux.
  - **SOCK\_DGRAM** Datagrames sense connexió.
  - **SOCK\_RAW** Per accedir directament a la interfície.
- **protocol:** Protocol específic per la transmissió. En el nostre cas **IPPROTO\_TCP**.

Exemple:

```
#include <sys/socket.h>

int s = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP);
```

### ● **Estructura sockaddr\_in**

Inclosa en el header **netinet/in.h**

```
struct sockaddr_in{

    short sin_family;           //Família d'adreces

    unsigned short sin_port;    //Número de port

    struct in_addr sin_addr;    //Adreça d'internet

};
```

on

```
struct in_addr {

    unsigned long s_addr; };
```

## Exemple

```
struct sockaddr_in origen;  
origen.sin_family = AF_INET;  
origen.sin_port = htonl(12345);  
//origen.sin_port = htons(0);           //port a l'atzar.  
origen.sin_addr.s_addr = INADDR_ANY;    // Que l'operatiu triï l'adreça.
```

Per guardar en el camp `s_addr` una adreça, per exemple, `deic-dc10.uab.es`, caldrà utilitzar la funció:

```
int getaddrinfo(const char *node, const char *service,  
                const struct addrinfo *hints,  
                struct addrinfo **res);
```

On:

**const char \*node**: El nom del host.

Vegeu com s'utilitza en el codi esquelet.

- **int bind(int sockfd, struct sockaddr \*my\_addr, socklen\_t addrlen)**

Per associar un socket a una adreça: host + port.

- **sockfd**: Socket que volem associar a una adreça
- **my\_addr**: Adreça a la que volem associar el socket
- **addrlen**: Longitud de l'estructura de l'adreça anterior.

Exemple:

```
#include <sys/socket.h>  
.....  
bind(s, (struct sockaddr*)&origen, sizeof(origen));
```

- **int connect(int sockfd, const struct sockaddr \*addr, socklen\_t addrlen)**

Aquesta crida connecta el socket **sockfd** amb l'adreça **addr**.

- **int listen(int sockfd, int backlog)**

Aquesta crida li indica al sistema operatiu que el socket **sockfd** és un socket passiu, és a dir que espera rebre peticions de connexions.

- **int accept(int sockfd, struct sockaddr \*addr, socklen\_t \*addrlen)**

Aquesta crida fa les següents accions:

1. Accepta la connexió rebuda a través del socket **sockfd**.
2. Omple la variable **addr** amb les dades de la connexió remota, és a dir del client. Noteu que aquesta variable es passa per referència.
3. Omple, també per referència, el tamany de l'adreça que ha omplert en l'apartat anterior.

#### 4. Retorna un nou socket per utilitzar-lo en l'enviament/recepció de missatges amb el client.

Exemple:

```
#include <sys/socket.h>
.....
struct sockaddr_in client_addr;
socklen_t client_addrlen = sizeof(client_addr); //SUPER IMPORTANT
int s = socket(...);
int s2;
s2 = accept(s, (struct sockaddr *)&client_addr, &client_addrlen);
```

#### • ssize\_t send(int sockfd, const void \*buf, size\_t len, int flags);

Enviament de dades per un socket amb connexió.

- S: Socket per on enviarem el buffer de dades
- buf: Buffer amb les dades a enviar
- len: Longitud del buffer amb les dades a enviar.
- flags: Pel control de l'enviament. Normalment pren el valor 0.

Exemple:

```
#include <sys/socket.h>
.....
send(s, missatge, long_missatge, 0);
```

#### • int recv(int s, void \*buf, size\_t lon, int flags )

Reb dades d'un socket amb connexió. És blocant.

- s: Socket per on rebem dades
- buf: Buffer per omplir amb les dades rebudes.
- lon: Longitud del buffer de dades rebudes
- flags: Pel control de la recepció, p.e: opció per poder veure les dades rebudes sense treure-les del buffer... Normalment 0.

Exemple:

```
#include <sys/socket.h>
.....
int bytes_rebut = recv(s_escolta, peticio, sizeof(peticio), 0);
```

- **uint16\_t ntohs(uint16\_t netshort);**

Converteix el paràmetre 'netshort' de *network byte order* a *host byte order*.

- **uint16\_t htons(uint16\_t hostshort);**

Converteix el paràmetre 'hostshort' de *host byte order* a *network byte order*.

- **uint16\_t ntohl(uint32\_t netlong);**

Converteix el paràmetre 'netlong' de *network byte order* a *host byte order*.

- **uint16\_t htonl(uint32\_t hostlong);**

Converteix el paràmetre 'hostlong' de *host byte order* a *network byte order*.

## Llibreríes que us calen importar

Us calen importar les següents llibreries per poder carregar totes les *system calls* anteriors:

```
#include <unistd.h> //per tipus com el socklen_t
#include <netdb.h> //pel gethostbyname
#include <errno.h> //per gestionar errors
#include <sys/types.h> //per tipus com el uint
#include <netinet/in.h> //per sockaddr_in
#include <sys/socket.h> //per la creació de sockets
```

## Diagrama de flux

