

INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO
SISTEMAS OPERATIVOS
GRUPO: 2CM8

ALUMNO:
NAVA VIVAS ANA PAOLA

CHAT

La práctica consiste en crear un chat a partir del uso de hilos, socket, semáforos y sincronización de procesos.

Cliente:

```
    printf("Uso: %s <puerto>\n",argv[0]);  
    exit(1);  
}  
if((id_socket=socket(AF_INET,SOCK_STREAM,0))==-1)  
{  
    printf("\n error al crear el socket\n");  
    exit(1);  
}  
printf("\nSocket creado\n");  
memset((void *)&servidor,0,sizeof(servidor));  
servidor.sin_family=AF_INET; //la misma familia del socket  
servidor.sin_port=htons(atoi(argv[1])); //transforma el puerto de string a int y luego a host  
servidor.sin_addr.s_addr=INADDR_ANY;  
//paso 2 Hacer la publicacion del socket  
if(bind(id_socket,(struct sockaddr *)&servidor,(socklen_t)sizeof(servidor))==-1)  
{  
    printf("\nerror al hacer la publicacion del socket\n");  
    close(id_socket);  
    exit(1);  
}  
printf("\nsocket abierto\n");  
//paso 3 ponerse a escuchar peticiones  
if(listen(id_socket,BACKLOG)==-1) //el 5 esta casi de mas xdxdx
```

El cliente se conecta al servidor y comienza a mandar mensajes. El cliente envía el mensaje al servidor y el servidor lo replica a otros clientes conectados al mismo servidor.

Servidor:

El servidor “resuelve la petición del cliente y busca los que le pide para así retornarlos. Se puede pensar como un proceso de categorización de mensajes (para que cada cliente no reciba el mensaje que envió).

```
^C
kolt@debian CHAT $ sudo rm /dev/shm/*
kolt@debian CHAT $ ./cli 5555 127.0.0.1

Socket creado

error al conectarse al socket
kolt@debian CHAT $ ./cli 5556 127.0.0.1

Socket creado

Se ha conectado con el servidor con socket 3

Mi IP: 127.0.0.1
Guest: hola
Guest:                (OK)
qué onda
Guest:                (OK)
heu
Guest:                (OK)
no mucho y tú?
pues aquí
Guest:                (OK)
```

```
^C
kolt@debian CHAT $ sudo rm /dev/shm/*
kolt@debian CHAT $ ./cli 5555 127.0.0.1

Socket creado

error al conectarse al socket
kolt@debian CHAT $ ./cli 5556 127.0.0.1

Socket creado

Se ha conectado con el servidor con socket 3

Mi IP: 127.0.0.1
Guest: hola
Guest:                (OK)
qué onda
Guest:                (OK)
heu
Guest:                (OK)
no mucho y tú?
pues aquí
Guest:                (OK)
```

CONCLUSIÓN:

El servidor se encarga de monitorear a todos sus clientes, desconectándolos al recibir una orden de cierre de canal. Vemos cómo se pueden comunicar las personas a través de un sencillo sistema de chat utilizando los protocolos tan conocidos como TCP e IP.