



SISTEMAS OPERATIVOS 1

MAESTRO: JOSUE PEREZ – LUCERO

ALUMNO: IRVYN XICALE CABRERA 201963582

CÓDIGO

servidor.c

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/socket.h>
#include <netdb.h>
#include <netinet/in.h>
#include <sys/types.h>
#include <string.h>

int main(int argc, char *argv[]){
    int sock, tam, n, numClientes = 0;
    struct sockaddr_in origenTam[5], origenTemp;
    struct sockaddr_in origen[5], origenTemp;
    struct sockaddr_in servidor;
    char buffer[1024];
    char usuario[10];
    if(argc < 2){
        printf("Error en la cantidad de parametros, se requiere establecer un puerto \n");
        exit(0);
        return 0;
    }
    sock = socket(AF_INET, SOCK_DGRAM, 0);
    tam = sizeof(servidor);
    bzero(&servidor, tam);
    servidor.sin_family = AF_INET;
    servidor.sin_addr.s_addr = INADDR_ANY;
    servidor.sin_port = htons(atoi(argv[1]));
    if(bind(sock, (struct sockaddr *)&servidor, tam) < 0){
        printf("Error de Conexion \n");
        return 0;
    }
    origenTamTemp = sizeof(struct sockaddr_in);
    printf("SERVIDOR ACTIVO ----- ESPERA A RECIBIR MENSAJES-----\n");
    while(1){
        strcpy(usuario, "");
        strcpy(buffer, "");
        int u = recvfrom(sock, usuario, 10, 0, (struct sockaddr *)
        &origen[numClientes], &origenTam[numClientes]);
        if(u < 0){
            printf("error recibir datos \n");
            exit(0);
        }else{
            n = recvfrom(sock, buffer, 1024, 0, (struct sockaddr *)
            &origen[numClientes], &origenTam[numClientes]);
```

```

}
if(numClientes != 3){
numClientes++;
}

write(1,usuario,u);
write(1,": ",2);
write(1,buffer,n);
for (int i = 0; i < numClientes; i++)
{
n = sendto(sock,usuario,strlen(usuario),0, (struct sockaddr *)&origen[i], origenTam[i]);
u = sendto(sock,buffer,strlen(buffer),0, (struct sockaddr *)&origen[i], origenTam[i]);
if(n < 0){
printf("error al enviar datos %d, %d\n",numClientes,i);
exit(0);
} else{
printf("buffer: %d, usuario: %d",n,u);
}
}
strcpy(usuario,"");
strcpy(buffer,"");
}
return 0;
}

```

clientes.c

```

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/socket.h>
#include <netdb.h>
#include <netinet/in.h>
#include <sys/types.h>
#include <string.h>

int main(int argc, char *argv[]){
pid_t pA;
int sock, tam, n, u, enviado;
socklen_t origenTam;
struct hostent *huesped;
struct sockaddr_in origen, servidor;
char buffer[1024];
char usuario[10];
if(argc < 4){
printf("Error en la cantidad de parametros, se requiere establecer un puerto \n");

```

```
printf("Sintaxis: ./prog servidor puerto usuario(maximo 10 caracteres)\n");
exit(0);
return 0;
}
```

```
sock = socket(AF_INET, SOCK_DGRAM, 0);
tam = sizeof(struct sockaddr_in);
bzero(&servidor,tam);
huesped = gethostbyname(argv[1]);
if( huesped <= 0){
printf("El servidor no existe\n");
return 0;
}
bzero(buffer,1024);
bcopy((char *)huesped->h_addr,(char *)&servidor.sin_addr,huesped->h_length);
servidor.sin_family = AF_INET;
servidor.sin_addr.s_addr = INADDR_ANY;
servidor.sin_port = htons(atoi(argv[2]));
```

```
char entrada[1024];
char temp[1024];
int cmp = 1;
while(cmp != 10 ){
//printf("Ingresa una cadena: \n");
strcpy(usuario,"");
strcpy(buffer,"");
fgets(buffer,1024,stdin);
//printf("%s",temp);
```

```
cmp = strcmp(entrada,"adios");
//printf("Buffer: %sEntrada:%s CMP: %d \n",buffer,entrada,cmp);
n = sendto(sock, argv[3],strlen(argv[3]),0, (struct sockaddr *)&servidor, tam);
if(n < 0){
printf("error al enviar el mensaje \n");
exit(0);
}
n = sendto(sock, buffer,strlen(buffer),0, (struct sockaddr *)&servidor, tam);
if(n < 0){
printf("error al enviar el mensaje \n");
exit(0);
}
if(((pA=fork()))==0){
while (1)
{
origenTam = sizeof(struct sockaddr_in);
u = recvfrom(sock,usuario,10,0,(struct sockaddr *)&origen,&tam);
if(u < 0){
```

```

printf("error recibir datos \n");
exit(0);
}
n = recvfrom(sock,buffer,1024,0,(struct sockaddr *)&origen,&tam);
if(n < 0){
printf("error recibir datos \n");
exit(0);
}
write(1,usuario,u);
write(1, ": ",2);
write(1,buffer,n);
}
}
}
close(sock);
return 0;
}

```

SERVIDOR

```

irvyn@irvyn-System-Product-Name:~/Documentos/S01/socket$ ./servidor 1200
SERVIDOR ACTIVO ----- ESPERA A RECIBIR MENSAJES-----
usuario01: hola
usuario02: hola
usuario02: ¿como estas?
usuario01: muy bien :)

```

Lín. 88, Col. 1

USUARIO02

```

irvyn@irvyn-System-Product-Name:~/Documentos/S01/socket$ ./clientes localhost 1200 usuario02
hola
usuario02: hola
¿como estas?
usuario02: ¿como estas?
usuario02: ¿como estas?
usuario01: muy bien :)
?
usuario01: muy bien :)

```

USUARIO01

```
irvyn@irvyn-System-Product-Name:~/Documentos/S01/socket$ ./clientes localhost 1200 usuario01
hola
usuario01: hola
usuario02: hola
usuario02: ¿como estas?
muy bien :)
usuario01: muy bien :)
```