

## Reporte del Capitulo 10: Cliente Internet tipo UDP

### Pregunta 10-1

`inet_addr("127.0.0.1")`

Esta función se encuentra en la librería `arpa/inet.h`

Esta función convierte la cadena a la que apunta, en el estándar IPv4 en notación decimal, a un valor entero adecuado para su uso como una dirección de Internet.

Y la dirección 127.0.0.1 se debe a que es la dirección localhost de nuestra pc, o sea, la dirección que apunta a nuestra PC desde nuestra PC, y se le llama la dirección IP de loopback. El loopback se refiere al enrutamiento del flujo de datos.

### Código del Servidor UDP en C:

```
#include <sys/types.h>
#include <sys/socket.h>
#include <stdio.h>
#include <netinet/in.h>
#include <netdb.h>
#include <strings.h>

int puerto = 7200;

int main(void)
{
    int num[2];
    int s, res, clilen;
    struct sockaddr_in server_addr, msg_to_client_addr;
    int tam, puerto_cliente;
    unsigned char ip_servidor[4], ip_cliente[4];

    s = socket(AF_INET, SOCK_DGRAM, 0);

    /* se asigna una direccion al socket del servidor*/
    bzero((char *)&server_addr, sizeof(server_addr));
    server_addr.sin_family = AF_INET;
    server_addr.sin_addr.s_addr = INADDR_ANY;
    server_addr.sin_port = htons(puerto);
    bind(s, (struct sockaddr *)&server_addr, sizeof(server_addr));
    clilen = sizeof(msg_to_client_addr);
    memcpy(ip_servidor, &server_addr.sin_addr.s_addr+0,4);
    memcpy(ip_cliente, &msg_to_client_addr.sin_addr.s_addr+0,4);

    printf("\nPuerto local: %d", server_addr.sin_port);
    while(1)
    {
```

```

    tam = recvfrom(s, (char *) num, 2*sizeof(int), 0, (struct sockaddr *)&msg_to_client_addr,
&clilen);
    res = num[0] + num[1];

    if(tam == -1)
    {
        perror("\nError al recibir");
        exit(0);
    }
    else
    {
        printf("\n\nIP del cliente: ");
        for(int i=0; i<4; i++)
            printf("%d. ", ip_cliente[i]);
        printf("\nPuerto del cliente: %d", ntohs(msg_to_client_addr.sin_port));
    }
    /* envía la petición al cliente. La estructura msg_to_client_addr contiene la dirección socket del
cliente */
    sendto(s, (char *)&res, sizeof(int), 0, (struct sockaddr *)&msg_to_client_addr, clilen);
}
}

```

## Código del Cliente UDP en C

```

#include <sys/types.h>
#include <sys/socket.h>
#include <stdio.h>
#include <netinet/in.h>
#include <netdb.h>
#include <strings.h>
#include <arpa/inet.h>

int puerto = 7200;

int main(int argc, char const *argv[])
{
    struct sockaddr_in msg_to_server_addr, client_addr;
    int s, num[2], res;
    int tam, puerto_cliente;
    unsigned char ip_cliente[4], ip_servidor[4];

    puerto_cliente = atoi(argv[2]);
    s = socket(AF_INET, SOCK_DGRAM, 0);
    /* rellena la dirección del servidor */
    bzero((char *)&msg_to_server_addr, sizeof(msg_to_server_addr));

```

```
msg_to_server_addr.sin_family = AF_INET;
msg_to_server_addr.sin_addr.s_addr = inet_addr(argv[1]);
msg_to_server_addr.sin_port = htons(puerto_cliente);

/* rellena la direcciòn del cliente*/
bzero((char *)&client_addr, sizeof(client_addr));
client_addr.sin_family = AF_INET;
client_addr.sin_addr.s_addr = INADDR_ANY;
client_addr.sin_port = htons(puerto_cliente);

memcpy(ip_servidor, &msg_to_server_addr.sin_addr.s_addr+0,4);
memcpy(ip_cliente, &client_addr.sin_addr.s_addr+0,4);

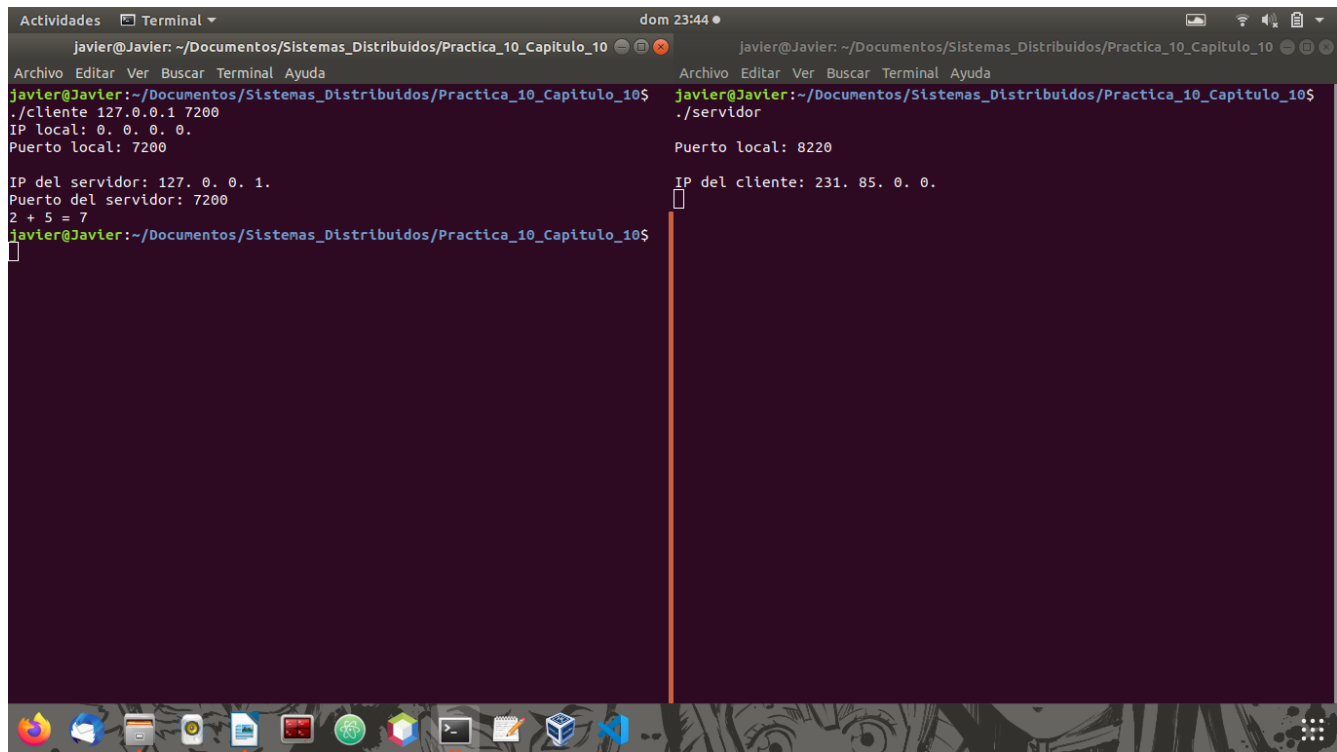
bind(s, (struct sockaddr *)&client_addr,sizeof(client_addr));
num[0] = 2;
num[1] = 5; /*rellena el mensaje */
sendto(s, (char *)num, 2 * sizeof(int), 0, (struct sockaddr *) &msg_to_server_addr,
sizeof(msg_to_server_addr));

printf("IP local: ");
    for(int i=0; i<4; i++)
        printf("%d. ", ip_cliente[i]);
printf("\nPuerto local: %d\n", ntohs(client_addr.sin_port));

tam = recvfrom(s, (char *)&res, sizeof(int), 0, NULL, NULL);
if(tam == -1)
{
    perror("\nError al recibir.\n");
    exit(0);
}
else
{
    printf("\nIP del servidor: ");
        for(int i=0; i<4; i++)
            printf("%d. ", ip_servidor[i]);
    printf("\nPuerto del servidor: %d\n", htons(msg_to_server_addr.sin_port));
}
/* se bloquea esperando respuesta */

printf("2 + 5 = %d\n", res);
close(s);
}
```

## Captura de pantalla



```
dom 23:44 ●
javier@Javier: ~/Documentos/Sistemas_Distribuidos/Practica_10_Capitulo_10
Archivo Editar Ver Buscar Terminal Ayuda
javier@Javier:~/Documentos/Sistemas_Distribuidos/Practica_10_Capitulo_10$ ./cliente 127.0.0.1 7200
IP local: 0. 0. 0. 0.
Puerto local: 7200
IP del servidor: 127. 0. 0. 1.
Puerto del servidor: 7200
2 + 5 = 7
javier@Javier:~/Documentos/Sistemas_Distribuidos/Practica_10_Capitulo_10$

javier@Javier: ~/Documentos/Sistemas_Distribuidos/Practica_10_Capitulo_10
Archivo Editar Ver Buscar Terminal Ayuda
javier@Javier:~/Documentos/Sistemas_Distribuidos/Practica_10_Capitulo_10$ ./servidor
Puerto local: 8220
IP del cliente: 231. 85. 0. 0.
```