INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

Desarrollo de Sistemas Distribuidos

Ukranio Coronilla Contreras

Capítulo No. 10

CLIENTE INTERNET TIPO UDP

EQUIPO 2

Cervantes Varela Juan Manuel

Martell Fuentes Ambar Desiree

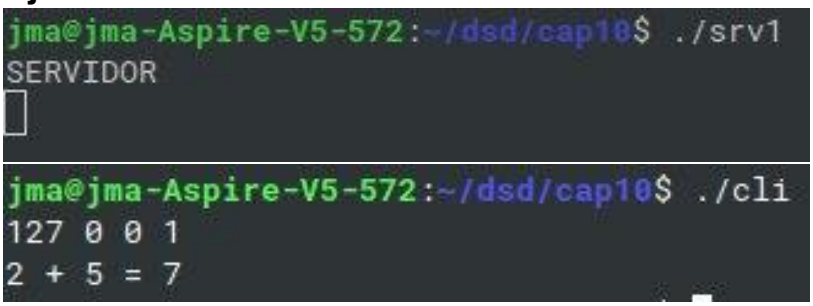
4CM4

Investigue cual es la función que realiza inet_addr(“127.0.0.1”) en el programa 10-1. ¿Por qué se la pasa la dirección 127.0.0.1 como argumento?

La función inet_addr () convierte la dirección de host de Internet de la notación de números y puntos IPv4 en datos binarios en orden de bytes de red.

Se usa la dirección 127.0.0.1 porque es la dirección de localhost, para que apunte a la misma pc desde donde se ejecuta el socket.

Ejercicio 10.1



Ejercicio 10.2

```
jma@jma-Aspire-V5-572:~/dsd/cap10$ ./srv1
SERVIDOR
IP address: 127.0.0.1
Port: 49795
IP address: 127.0.0.1
Port: 49647
IP address: 127.0.0.1
Port: 47195
IP address: 127.0.0.1
Port: 34101
```

```
jma@jma-Aspire-V5-572:~/dsd/cap10$ ./cli
127 0 0 1
IP address: 127.0.0.1
Port: 7200
2 + 5 = 7
jma@jma-Aspire-V5-572:~/dsd/cap10$ ./cli
127 0 0 1
IP address: 127.0.0.1
Port: 7200
2 + 5 = 7
jma@jma-Aspire-V5-572:~/dsd/cap10$ ./cli
127 0 0 1
IP address: 127.0.0.1
Port: 7200
2 + 5 = 7
jma@jma-Aspire-V5-572:~/dsd/cap10$ ./cli
127 0 0 1
IP address: 127.0.0.1
Port: 7200
2 + 5 = 7
jma@jma-Aspire-V5-572:~/dsd/cap10$
```

Ejercicio 10.3

```
jma@jma-Aspire-V5-572:~/dsd/cap10$ ./srv1
SERVIDOR
IP address: 192.168.1.73
Port: 48387
IP address: 192.168.1.73
Port: 52337
IP address: 192.168.1.73
Port: 33194
IP address: 192.168.1.73
Port: 42897
```

```
vagrant@buster:~/dsd$ ./cli 192.168.1.75
192 168 1 75
IP address: 192.168.1.75
Port: 7200
2 + 5 = 7
vagrant@buster:~/dsd$ ./cli 192.168.1.75
192 168 1 75
IP address: 192.168.1.75
Port: 7200
2 + 5 = 7
vagrant@buster:~/dsd$ ./cli 192.168.1.75
192 168 1 75
IP address: 192.168.1.75
Port: 7200
2 + 5 = 7
vagrant@buster:~/dsd$ ./cli 192.168.1.75
192 168 1 75
IP address: 192.168.1.75
Port: 7200
2 + 5 = 7
vagrant@buster:~/dsd$
```

Código fuente

Servidor

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

  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 = inet_addr("192.168.1.75");

        // printf("\n Tamanio de puerto: %d\n", (int)sizeof(puerto));
        // printf("\n Valor de puerto: %x\n", puerto);
        // printf("\n Antes de htons: %x\n", server_addr.sin_port);

  server_addr.sin_port = htons(puerto);

        // printf("\n Despues de htons: %x\n", server_addr.sin_port);

  bind(s, (struct sockaddr *)&server_addr, sizeof(server_addr));
  printf("SERVIDOR\n");
  clilen = sizeof(msg_to_client_addr);
  while(1) {
    recvfrom(s, (char *) num, 2*sizeof(int), 0, (struct sockaddr
*)&msg_to_client_addr, &clilen);
    char *client = inet_ntoa(msg_to_client_addr.sin_addr);
    printf("IP address: %s\n", client);
        printf("Port: %i\n", ntohs(msg_to_client_addr.sin_port));

    res = num[0] + num[1];
```

```
    /* 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);
  }
}
```

Cliente

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

using namespace std;

int puerto = 7200;

int main(int argc, char* argv[]) {
  // 10_3
  if (argc<=1) {
    printf("\n\tNo se ingreso direccion IP de servidor.\n\n");
    return 1;
  }

  struct sockaddr_in msg_to_server_addr, client_addr;
  int s, num[2], res;

  s = socket(AF_INET, SOCK_DGRAM, 0);

  /* rellena la direccion del servidor */
  bzero((char*)&msg_to_server_addr, sizeof(msg_to_server_addr));
  msg_to_server_addr.sin_family = AF_INET;
  // 10_3
  msg_to_server_addr.sin_addr.s_addr = inet_addr(argv[1]);
  // msg_to_server_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
  msg_to_server_addr.sin_port = htons(puerto);

  // 10_1
  //printf("%u\n", msg_to_server_addr.sin_addr.s_addr);
  unsigned char aux[4];
  memcpy(aux, (unsigned char*)&msg_to_server_addr.sin_addr.s_addr, 4);
```

```
    for(int i=0 ; i<4 ; i++)
        printf("%u ", aux[i]);
    printf("\n");

    /* rellena la direccion del cliente */
    bzero((char*)&client_addr, sizeof(client_addr));
    client_addr.sin_family = AF_INET;
    client_addr.sin_addr.s_addr = INADDR_ANY;
    /* cuando se utiliza por numero de puerto el 0, el sistema se encarga de
asignarle uno */
    client_addr.sin_port = htons(0);

    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));

    /* se bloquea esperando la respuesta */
    recvfrom(s, (char*)&res, sizeof(int), 0, NULL, NULL);

    // 10_2
    char *server = inet_ntoa(msg_to_server_addr.sin_addr);
    printf("IP address: %s\n", server);
          printf("Port: %i\n", ntohs(msg_to_server_addr.sin_port));

    printf("2 + 5 = %d\n", res);

    close(s);
}
```