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 cunción 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

