```
/*    Luis C. Aparicio

        Sistemas Operativos
        EUPT: Área de Arquitectura y Tecnología de computadores

   Programa cliente para la suma utilizando sockets datagram (UDP)

   clieUDP.c      */

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

int main(int argc, char *argv[])
{
  struct sockaddr_in server_addr, client_addr;
  struct hostent *hp;

  char *namehost;
  int sd, puerto, res;
  int num[2];

  namehost = argv[1];         /* Tratamiento de parámetros */
  puerto = atoi(argv[2]);
  num[0] = atoi(argv[3]);
  num[1] = atoi(argv[4]);

  sd = socket(AF_INET, SOCK_DGRAM, 0);

  /* Se optine y se rellena la dirección del servidor */
  bzero((char *)&server_addr, sizeof(server_addr));

  hp = gethostbyname(namehost);
  if (hp == NULL)
  {
    printf("\n Error en la llamada gethostbyname");
    exit(0);
  }

  bcopy (hp->h_addr, (char *)&(server_addr.sin_addr), hp->h_length);
  server_addr.sin_family = AF_INET;
  server_addr.sin_port = htons(puerto);

  /* Se rellena la dirección del cliente, cuando se utiliza el número de puerto
0,
     el sistema se encarga de asignarle uno */
  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(0); /*puerto asignado por el sistema*/

  bind(sd, (struct sockaddr *)&client_addr, sizeof(client_addr));

  sendto(sd, num, 2 * sizeof(int), 0, (struct sockaddr *)&server_addr,
sizeof(server_addr));
  recvfrom(sd, &res, sizeof(int), 0, NULL, NULL);

  printf("\n El resultado es %d \n", res);
```



```
  close(sd);
  exit(0);
}
```

```
/*
   Luis C. Aparicio

        Sistemas Operativos
        EUPT: Área de Arquitectura y Tecnología de computadores

   Programa servidor de suma utilizando sockets datagram (UDP)

   servUDP.c
*/

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

int main()
{
  struct sockaddr_in server_addr, client_addr;

  int sd;
  int size, res;
  int num[2];

  sd = socket(AF_INET, SOCK_DGRAM, 0);

  /* Asigna la dirección al socket */
  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(4400);

  bind(sd, (struct sockaddr *)&server_addr, sizeof(server_addr));

  size = sizeof(client_addr);

  while (1)
  {
    printf("\nEsperando conexión en el puerto: 4400");

    //recvfrom (sd, (char*) num, 2 * sizeof(int), 0, (struct sockaddr
*)&client_addr, &size);
    recvfrom (sd, num, 2 * sizeof(int), 0, (struct sockaddr *)&client_addr,
&size);

    res = num[0] + num[1];
    printf("\nServicio: %d + %d = %d \n", num[0], num[1], res);

    /* Envía la petición al cliente.
       La estructura client_addr contiene la dirección del socket del cliente */

    //sendto(sd, (char*)&res, sizeof(int), 0, (struct sockaddr *)&client_addr,
size);
    sendto(sd, &res, sizeof(int), 0, (struct sockaddr *)&client_addr, size);
  }
}
```

```
/*   Luis C. Aparicio

        Sistemas Operativos
        Área de Arquitectura y Tecnología de computadores

   Programa cliente para la suma utilizando sockets stream (TCP)

   clieTCP.c      */

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

int main(int argc, char *argv[])
{
  struct sockaddr_in server_addr;
  struct hostent *hp;

  char *namehost;
  int sd, puerto, res;
  int num[2];

  namehost = argv[1];         /* Tratamiento de parámetros */
  puerto = atoi(argv[2]);
  num[0] = atoi(argv[3]);
  num[1] = atoi(argv[4]);

  sd = socket(AF_INET, SOCK_STREAM, 0);

  /* Se obtine y se rellena la dirección del servidor */
  bzero((char *)&server_addr, sizeof(server_addr));

  /* hp = gethostbyname("esquinazo.unizar.es"); */
  hp = gethostbyname(namehost);

  if (hp == NULL)
  {
    printf("\n Error en la llamada gethostbyname");
    exit(0);
  }

  bcopy (hp->h_addr, (char *)&(server_addr.sin_addr), hp->h_length);
  server_addr.sin_family = AF_INET;
  server_addr.sin_port = htons(puerto);

  /* Se establece la conexión */
  if (connect(sd, (struct sockaddr *) &server_addr, sizeof(server_addr)) < 0)
  {
    perror("\n Error en la llamada connect");
    exit(0);
  }

    /* envía la petición */
  write(sd, &num, 2 *sizeof(int));

    /* recibe la respuesta */
  read(sd, &res, sizeof(int));

  printf("\n El resultado es %d \n", res);
```



```
  close(sd);

  exit(0);
}
```

```
/*
   Luis C. Aparicio

        Sistemas Operativos
        Área de Arquitectura y Tecnología de computadores

   Programa servidor de suma utilizando sockets stream (TCP)

   servTCP.c
*/

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

int main()
{
  struct sockaddr_in server_addr, client_addr;

  int sd, sc, salir;
  int size, res;
  int num[2];

  sd = socket(AF_INET, SOCK_STREAM, 0);

  /* Asigna la dirección al socket */
  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(4200);

  bind(sd, &server_addr, sizeof(server_addr));

  listen(sd, 5);

  size = sizeof(client_addr);
  salir = 0;

  while (salir == 0)
  {
    printf("Esperando conexión en el puerto: 4200\n");
    sc = accept(sd, (struct sockaddr *)&client_addr, &size);
    if(sc < 0) salir = 1;
    else
    {
      /* recibe la petición, dos número enteros */
      //read(sc, (char*) num, 2*sizeof(int));
      read(sc, &num, 2*sizeof(int));
      res = num[0] + num[1];

      printf("\nServicio: %d + %d = %d \n", num[0], num[1], res);

      /* envía el resultado y cierra la conexión */
      //write(sc, (char*)&res, sizeof(int));
      write(sc, &res, sizeof(int));
      close (sc);
    }
  }

  printf("\n Error en accept \n");
```



```
  close(sd);

  exit(0);
}
```