

Tecnológico De Estudios Superiores De Jocotitlan

Ing. Sistemas Computacionales

Taller De Sistemas Operativos

Segunda Oportunidad

Practica Final Competencia 3 y 4

Alumno:

José Eduardo Hernández Tomas

Fecha

30-Junio-2021

Docente:

Fernando Mercado Salinas

Ic-601

La red cliente-servidor es una red de comunicaciones en la cual los clientes están conectados a un servidor, en el que se centralizan los diversos recursos y aplicaciones con que se cuenta; y que los pone a disposición de los clientes cada vez que estos son solicitados.

Cliente:

```

C:\Users\ASUS\Desktop\ESCUELA 1\SEXTO SEMESTRE\TALLER DE SISTEMAS OPERATIVOS\programas taller de so\competencia 3 y 4\practica final\cliente_S.c - Dev-C++ 5.11
Archivo Edición Buscar Ver Proyecto Ejecutar Herramientas AStyle Vegtana Aysda
(globales)
Proyecto Clases(Fun) [*] cliente_S.c [*] servidor_S.c
1 #include <unistd.h>
2 /* includes para los SOCKETS */
3 #include <netdb.h>
4 #include <netinet/in.h>
5 #include <sys/socket.h>
6 #include <arpa/inet.h>
7 /*Para errores*/
8 #include <errn.h>
9 #include <stdio.h>
10 #include <stdlib.h>
11 #include <string.h>
12
13 #define DATO_MAXIMO 10000 //definimos el dato maximo
14 #define PORT 8080 //Definimos el puerto 8080 que es la alternativa para usar el http
15 #define Sock struct sockaddr
16
17 //funcion para pedir numeros
18 void pedir_Numeros (char numeros[]){
19     printf ("Ingresa el número: ");
20     scanf ("%d", numeros);
21 }
22
23
24 int main(int argc, char **argv)

```

```

C:\Users\ASUS\Desktop\ESCUELA 1\SEXTO SEMESTRE\TALLER DE SISTEMAS OPERATIVOS\programas taller de so\competencia 3 y 4\practica final\cliente_S.c - Dev-C++ 5.11
Archivo Edición Buscar Ver Proyecto Ejecutar Herramientas AStyle Vegtana Aysda
(globales)
Proyecto Clases(Fun) [*] cliente_S.c [*] servidor_S.c
25 {
26
27     if(argc<2){
28         printf("<HOST > <PUERTO>\n");
29         return 1;
30     }
31
32     //DEFINIMOS LAS VARIABLES
33     //escuchar, file descriptor
34     int sockfd, connfd;
35
36     //ESTRUCTURA PARA EL SERVIDOR Y EL CLIENTE
37     //en esta estructura se define el formato de la direccion
38     //familia, puerto y la direccion IPv4
39     struct sockaddr_in servaddr, cliente_1;
40
41     //CREACION DEL SOCKET
42     //dominio, tipo, protocolo
43     sockfd = socket(AF_INET, SOCK_STREAM, 0);
44     //si hay un error, devuelve -1
45     if (sockfd == -1)
46     {
47         printf(stderr, "ERROR FALLA LA CREACION DEL SOCKET\n");
48         exit (0)

```

```
C:\Users\ASUS\Desktop\ESCUELA 1\SEXTO SEMESTRE\TALLER DE SISTEMAS OPERATIVOS\programas taller de so\competencia 3 y 4\practica final\cliente_S.c - Dev-C++ 5.11
Archivo Edición Buscar Ver Proyecto Ejecutar Herramientas AStyle Veptana Ayuda
(globals)
Proyecto Clases(Fun) [*] cliente_S.c [*] servidor_S.c
47 printf(stderr, "ERROR FALLA LA CREACION DEL SOCKET\n");
48 exit (0)
49 }
50 else
51 printf ("-->SOCKET CREADO EXITOSAMENTE \n");
52
53 //crea la estructura address format
54 memset(&servaddr,0,sizeof(servaddr));
55
56 //asignar IP, Puerto y direccion IPv4
57 servaddr.sin_family = AF_INET;
58 servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
59 servaddr.sin_port = htons(atoi(argv[1])); //este valor ordena Los bits
60
61 //CONECTAMOS LOS SOCKETS-- CLIENTE && SERVIDOR
62 if (connect(sockfd, (Socket*)&servaddr, sizeof(servaddr)) != 0) {
63 printf("ERROR LA CONEXION FALLA\n");
64 return -1;
65 }
66 else
67 printf("-->CONEXION CON SERVIDOR EXITOSO\n");
68
69 fd funcion(sockfd); //LLAMADA A LA FUNCION DEL FILE DESCRIPTOR
70
Activar Windows
Ve a Configuración para activar Windows.
```

```
C:\Users\ASUS\Desktop\ESCUELA 1\SEXTO SEMESTRE\TALLER DE SISTEMAS OPERATIVOS\programas taller de so\competencia 3 y 4\practica final\cliente_S.c - Dev-C++ 5.11
Archivo Edición Buscar Ver Proyecto Ejecutar Herramientas AStyle Veptana Ayuda
(globals)
Proyecto Clases(Fun) [*] cliente_S.c [*] servidor_S.c
71 close(sockfd); //TERMINAMOS EL SOCKET
72 }
73
74 void fd_funcion(int sockfd)
75 {
76 char numero1[DATO_MAXIMO];
77 char numero2[DATO_MAXIMO];
78 char resultado[DATO_MAXIMO];
79 char buff[DATO_MAXIMO]; //definimos buffer para recepcion
80
81 //borramos el area de memoria dada con cero byte
82 bzero(buff, sizeof(buff));
83 bzero(numero1, sizeof(numero1));
84 bzero(numero2, sizeof(numero2));
85 bzero(resultado, sizeof(resultado));
86
87 //mandamos a llamar a la funcion para pedir Los numeros
88 pedir_Numeros(numero1);
89 //funcion write: el cliente envÃ-a datos al servidor
90 write(sockfd, numero1, sizeof(numero1));
91 pedir_Numeros(numero2);
92 write(sockfd, numero2, sizeof(numero2));
93
94 bzero(buff, sizeof(buff));
95
Activar Windows
Ve a Configuración para activar Windows.
```

```
C:\Users\ASUS\Desktop\ESCUELA 1\SEXTO SEMESTRE\TALLER DE SISTEMAS OPERATIVOS\programas taller de so\competencia 3 y 4\practica final\cliente_S.c - Dev-C++ 5.11
Archivo Edición Buscar Ver Proyecto Ejecutar Herramientas AStyle Veptana Ayuda
(globais)
Proyecto Clases(Fun) [*] cliente_S.c [*] servidor_S.c
79 char buff[DATO_MAXIMO]; //definimos buffer para recepcion
80
81 //borramos el area de memoria dada con cero byte
82 bzero(buff, sizeof(buff));
83 bzero(numero1, sizeof(numero1));
84 bzero(numero2, sizeof(numero2));
85 bzero(resultado, sizeof(resultado));
86
87 //mandamos a llamar a la funcion para pedir Los numeros
88 pedir_Numeros(numero1);
89 //funcion write: el cliente envÃ-a datos al servidor
90 write(sockfd, numero1, sizeof(numero1));
91 pedir_Numeros(numero2);
92 write(sockfd, numero2, sizeof(numero2));
93
94 bzero(buff, sizeof(buff));
95 bzero(resultado, sizeof(resultado));
96
97 //funcion read: Lee Lo que el servidor responde
98 //Lee el resultado de la suma
99 read(sockfd, resultado, sizeof(resultado));
100 printf("\n ***RESULTADO= %s (Longitud %d digitos)\n", resultado, strlen(resultado));
101 }
```

Activar Windows
Ve a Configuración para activar Windows.

Line: 12 Col: 1 Sel: 0 Lines: 101 Length: 2841 Insertar Done parsing in 0.047 seconds

Escribe aquí para buscar

04:54 p. m. 30/06/2021

Servidor:

```
C:\Users\ASUS\Desktop\ESCUELA 1\SEXTO SEMESTRE\TALLER DE SISTEMAS OPERATIVOS\programas taller de so\competencia 3 y 4\practica final\servidor_S.c - Dev-C++ 5.11
Archivo Edición Buscar Ver Proyecto Ejecutar Herramientas AStyle Veptana Ayuda
(globais)
Proyecto Clases(Fun) [*] cliente_S.c [*] servidor_S.c
1 #include <unistd.h>
2 /* includes para los SOCKETS */
3 #include <netdb.h>
4 #include <netinet/in.h>
5 #include <sys/socket.h>
6 #include <arpa/inet.h>
7 /*Para errores*/
8 #include <errn.h>
9 #include <stdio.h>
10 #include <stdlib.h>
11 #include <string.h>
12
13 #define DATO_MAXIMO 10000 //definimos el dato maximo
14 #define PORT 8080 //Definimos el puerto 8080 que es la alternativa para usar el http
15 #define Sock struct sockaddr
16
17 /*Definimos las variables*/
18 char numero1[DATO_MAXIMO];
19 char numero2[DATO_MAXIMO];
20 char resultado[DATO_MAXIMO];
21 char buff_rx[DATO_MAXIMO]; //definimos buffer para recepcion
22
23
24 /*Creamos el método para la SUMA que vamos a ocupar*/
```

Activar Windows
Ve a Configuración para activar Windows.

Line: 158 Col: 57 Sel: 0 Lines: 160 Length: 4222 Insertar Done parsing in 0.047 seconds

Escribe aquí para buscar

04:55 p. m. 30/06/2021

```

C:\Users\ASUS\Desktop\ESCUELA 1\SEXTO SEMESTRE\TALLER DE SISTEMAS OPERATIVOS\programas taller de so\competencia 3 y 4\practica final\servidor_S.c - Dev-C++ 5.11
Archivo Edición Buscar Ver Proyecto Ejecutar Herramientas AStyle Vespina Ayuda
(globals)
Proyecto Clases(Fun) [*] cliente_S.c [*] servidor_S.c
24 /*Creamos el mÃtodo para La SUMA que vamos a ocupar*/
25 void SUMAR_NUMEROS(char numero1[], char numero2[], char resultado[]){
26 /*Definimos variables que nos ayudaran*/
27 int longitud_MAX, n1, n2, i, j, k;
28 /*Auxiliares para almacenar lo calculado posteriormente*/
29 int aux_N1[n1], aux_N2[n2], aux_LONG[longitud_MAX];
30
31 /*CALCULAMOS LA LONGITUD DE AMBAS CADENAS
32 strlen --> Devuelve la longitud de una cadena*/
33 n1 = strlen (numero1);
34 n2 = strlen (numero2);
35
36 //REALIZAMOS LA COMPARACIÓN DE LA LONGITUD DE AMBOS NÚMEROS
37 //PARA DETERMINAR LA MÁXIMA LONGITUD
38 if(n1>n2)
39 longitud_MAX=n1;
40 else
41 longitud_MAX=n2;
42
43 /*CONVERTIMOS LOS NÚMEROS DE CARACTER A ENTERO*/
44 /*Usamos el truco de restar del dÃgito el nÃmero '0'*/
45 for ( i=0; i<n1; i++){
46 aux_N1[i] =n1[i]-'0';
47 }

```

Activar Windows
Ve a Configuración para activar Windows.

Line: 158 Col: 57 Sel: 0 Lines: 160 Length: 4222 Insertar Done parsing in 0.047 seconds

Escribe aquí para buscar

04:56 p. m.
30/06/2021

```

C:\Users\ASUS\Desktop\ESCUELA 1\SEXTO SEMESTRE\TALLER DE SISTEMAS OPERATIVOS\programas taller de so\competencia 3 y 4\practica final\servidor_S.c - Dev-C++ 5.11
Archivo Edición Buscar Ver Proyecto Ejecutar Herramientas AStyle Vespina Ayuda
(globals)
Proyecto Clases(Fun) [*] cliente_S.c [*] servidor_S.c
47 }
48 for ( j=0; j<n2; j++){
49 aux_N2[j] =n2[j]-'0';
50 }
51
52 j = n2 - 1;
53 k = longitud_MAX-1;
54
55 for (i=n1-1; i>0 && j>0; i--)
56 {
57 aux_LONG[k] = aux_N1[i]+ aux_N2[j];
58 j--;
59 k--;
60 }
61 //reducimos los contadores
62 for (;i>=0; --i, --j){
63 aux_LONG = aux_N1[j];
64 }
65 for (;i>=0; --j, --k){
66 aux_LONG = aux_N2[k];
67 }
68
69 for (k = longitud_MAX-1; k>0; k--)
70 {
71 if (aux_LONG[k]>9)
72 {
73 aux_LONG[k]= aux_LONG[k]%10;
74 }

```

Activar Windows
Ve a Configuración para activar Windows.

Line: 158 Col: 57 Sel: 0 Lines: 160 Length: 4222 Insertar Done parsing in 0.047 seconds

Escribe aquí para buscar

04:56 p. m.
30/06/2021

```
C:\Users\ASUS\Desktop\ESCUELA 1\SEXTO SEMESTRE\TALLER DE SISTEMAS OPERATIVOS\programas taller de so\competencia 3 y 4\practica final\servidor_S.c - Dev-C++ 5.11
Archivo Edición Buscar Ver Proyecto Ejecutar Herramientas AStyle Veptana Ayuda
IDM-GCC 4.9.2 64-bit Release
(global.s)
Proyecto Clases(Fun) [*] cliente_S.c [*] servidor_S.c
73     aux_LONG[k]= aux_LONG[k]%10;
74     //aux_LONG[k-1]= aux_LONG[k-1]+1;
75     aux_LONG[k]++;
76 }
77 }
78 i=0;
79 for(k=0; k<longitud_MAX; k++)
80 {
81     if(aux_LONG[k]>9)
82     {
83         resultado[i]=(aux_LONG[k]-10)+'0';
84     }
85     else
86         resultado[i]=aux_LONG[k]+'0';
87     i++;
88 }
89
90 int main(int argc, char const *argv[])
91 {
92     //DEFINIMOS LAS VARIABLES
93     //escuchar, descriptor, longitud
94     int sockfd, connfd, long1;
95     //ESTRUCTURA PARA EL SERVIDOR Y EL CLIENTE
96     //en esta estructura se define el formato de la direccion
97     //familia, puerto y la direccion IPv4
98     struct sockaddr_in servaddr, cliente_i;
99     //CREACION DEL SOCKET
100     //dominio, tipo, protocolo
101     sockfd = socket(AF_INET, SOCK_STREAM, 0);
102     //si hay un error, devuelve -1
103     if (sockfd == -1)
104     {
105         printf(stderr, "ERROR FALLA LA CREACION DEL SOCKET\n");
106         exit (0);
107     }
108     else
109     {
110         printf ("-->SOCKET CREADO EXITOSAMENTE \n");
111     }
112     //crea la estructura saddress format
113     memset(&servaddr,0,sizeof(servaddr));
114     //asignar IP, Puerto y direccion IPv4
115     servaddr.sin_family = AF_INET;
116     servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
117     servaddr.sin_port = htons(atoi(argv[1])); //este valor ordena los bits
118     //ASIGNAMOS DIRECCION IP Y PUERTO
119     //file descriptor puntero a la estructura, el tamaño de la estructura
120     //SI EL No. ES MAYOR A 0, ASIGNAR CORRECTAMENTE
121     if ((bind(sockfd, (struct sockaddr *)&servaddr, sizeof(servaddr))) !=0)
122     {
123         printf(stderr, "ERROR FALLA LA ASIGNACION DE LA DIRECCION IP Y PUERTO\n");
124         exit (0);
125     }
126 }
```

```
C:\Users\ASUS\Desktop\ESCUELA 1\SEXTO SEMESTRE\TALLER DE SISTEMAS OPERATIVOS\programas taller de so\competencia 3 y 4\practica final\servidor_S.c - Dev-C++ 5.11
Archivo Edición Buscar Ver Proyecto Ejecutar Herramientas AStyle Veptana Ayuda
IDM-GCC 4.9.2 64-bit Release
(global.s)
Proyecto Clases(Fun) [*] cliente_S.c [*] servidor_S.c
99     //CREACION DEL SOCKET
100     //dominio, tipo, protocolo
101     sockfd = socket(AF_INET, SOCK_STREAM, 0);
102     //si hay un error, devuelve -1
103     if (sockfd == -1)
104     {
105         printf(stderr, "ERROR FALLA LA CREACION DEL SOCKET\n");
106         exit (0);
107     }
108     else
109     {
110         printf ("-->SOCKET CREADO EXITOSAMENTE \n");
111     }
112     //crea la estructura saddress format
113     memset(&servaddr,0,sizeof(servaddr));
114     //asignar IP, Puerto y direccion IPv4
115     servaddr.sin_family = AF_INET;
116     servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
117     servaddr.sin_port = htons(atoi(argv[1])); //este valor ordena los bits
118     //ASIGNAMOS DIRECCION IP Y PUERTO
119     //file descriptor puntero a la estructura, el tamaño de la estructura
120     //SI EL No. ES MAYOR A 0, ASIGNAR CORRECTAMENTE
121     if ((bind(sockfd, (struct sockaddr *)&servaddr, sizeof(servaddr))) !=0)
122     {
123         printf(stderr, "ERROR FALLA LA ASIGNACION DE LA DIRECCION IP Y PUERTO\n");
124         exit (0);
125     }
126 }
```

C:\Users\ASUS\Desktop\ESCUELA 1\SEXTO SEMESTRE\TALLER DE SISTEMAS OPERATIVOS\programas taller de so\competencia 3 y 4\practica final\servidor_S.c - Dev-C++ 5.11

Archivo Edición Buscar Ver Proyecto Ejecutar Herramientas AStyle Vegtana Ayuda

(globals)

Proyecto Clases(Fun) [*] cliente_S.c [*] servidor_S.c

```
134 //llamada al sistema listen
135 //5 corresponde al backlog No. maximo de conexiones pendientes
136 if ((listen(sockfd, 5)) != 0)
137 {
138     printf("ERROR LA LLAMADA AL SISTEMA FALLA.\n");
139     exit(0);
140 }
141 else
142     printf("-->LLAMADA AL SISTEMA EXITOSA\n");
143
144 //cargamos en la variable longitud el tamaño de la estructura para el address formato del cliente
145 long1 = sizeof(cliente_1);
146
147 //Aceptamos a los clientes
148 connfd = accept(sockfd, (Sock*)&cliente_1, &long);
149 if (connfd < 0) {
150     printf("ERROR EL SERVIDOR NO PUDO ACEPTAR AL CLIENTE \n");
151     exit(0);
152 }
153 else
154     printf("-->CLIENTE ACEPTADO CORRECTAMENTE\n");
155
156 printf("-----CONECTADO AL SERVIDOR-----\n");
157
158 fd_funcion(connfd); //mandamos a llamar a la funcion del file descriptor
159 close(sockfd); //CERRAR SOCKET
160 }
```

Activar Windows
Ve a Configuración para activar Windows.

Compilador Recursos Registro de Compilación Depuración Resultados

Line: 151 Col: 17 Sel: 0 Lines: 160 Length: 4222 Insertar Done parsing in 0.047 seconds

Escribe aquí para buscar

04:57 p. m. 30/06/2021