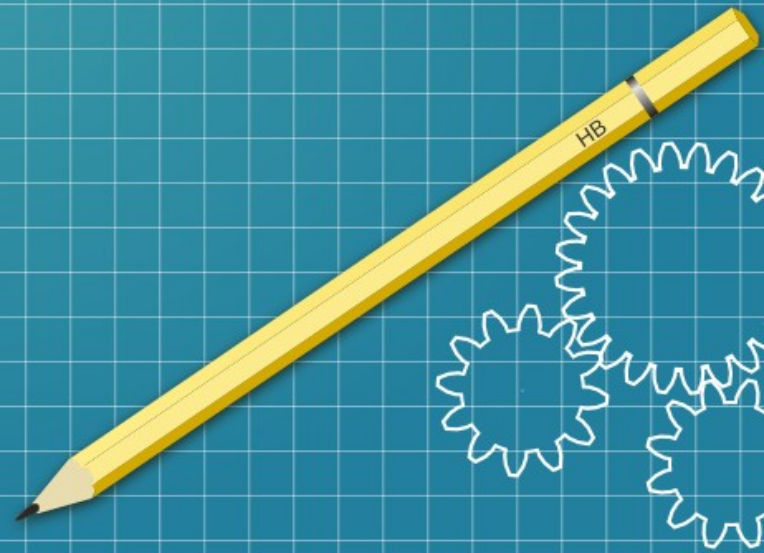


Cliente / Servidor Sockets

Gustavo Bacagine
Kaique Ryan



Modelo Cliente/Servidor

- Separa o programa em cliente e servidor como programas distintos para que se comunicam dentro da mesma rede



Bibliotecas e funções de socket em C



- `sys/socket.h`: Biblioteca C para manipulação de socket
- `socket(int domain, int type, int protocol)`: Cria um novo socket
- `bind(int socket, const struct sockaddr *address, socklen_t address_len)`: Associa o socket a um endereço
- `listen(int socket, int backlog)`: Coloca o socket no modo de escuta
- `accept(int socket, struct sockaddr *restrict address, socklen_t *restrict address_len)`: Aceita uma nova conexão
- `connect(int socket, const struct sockaddr *address, socklen_t address_len)`: Conecta o socket a um endereço

Exemplo em C - Servidor

```
1 /**
2  * server.c
3  *
4  * Developed by geek4geeks, modified by
5  * Gustavo Bacagine <gustavo.bacagine@protonmail.com>
6  *
7  * Example of server using sockets in C
8  *
9  * Compiling with: gcc -o server server.c -Wall -Wextra -O3 -ansi
10 *
11 * Date: 2023-10-31
12 */
13
14 /*****
15 *
16 *                               Includes
17 *
18 *****/
19 #include <stdio.h>
20 #include <stdbool.h>
21 #include <stdlib.h>
22 #include <string.h>
23 #include <strings.h> /* bzero() */
24 #include <unistd.h> /* read(), write(), close() */
25 #include <netdb.h>
26 #include <netinet/in.h>
27 #include <sys/socket.h>
28 #include <sys/types.h>
29
30 /*****
31 *
32 *                               Defines and macros
33 *
34 *****/
35
36 /**
37 * Used to ignore warnings in compilation
38 * about unused variables
39 */
40 #define UNUSED(X) (void) X
41
42 /**
43 * Communication definitions
44 */
```

NORMAL server.c

Exemplo em C - Servidor

```
45 #define MAX 80
46 #define PORT 8080
47
48 /*****
49 *
50 *           Typedefs, structures, unions and enums
51 *
52 *****/
53 typedef struct sockaddr STRUCT_SOCKADDR;
54 typedef struct sockaddr * PSTRUCT_SOCKADDR;
55
56 typedef struct sockaddr_in STRUCT_SOCKADDR_IN;
57 typedef STRUCT_SOCKADDR_IN * PSTRUCT_SOCKADDR_IN;
58
59 /*****
60 *
61 *           Functions
62 *
63 *****/
64
65 /**
66  * Function designed for chat between stClient and server.
67  */
68 void vChat(int iConnFd)
69 {
70     char szBuf[MAX];
71     int ii;
72
73     /**
74      * infinite loop for chat
75      */
76     while(true)
77     {
78         bzero(szBuf, MAX);
79
80         /**
81          * read the message from stClient and copy it in szBufer
82          */
83         read(iConnFd, szBuf, sizeof(szBuf));
84
85         /**
86          * print szBufer which contains the stClient contents
87          */
88         printf("From Clientent: %s"
```

NORMAL server.c

c utf-8[unix]

21% 1:45/209= %:1

[4]trailing [106]mixed-indent [70:74]mix-indent-file

Exemplo em C - Servidor

```
89 | "Type a string to send to the client: ", szBuf
90 | );
91 | bzero(szBuf, MAX);
92 | ii = 0;
93 |
94 | /**
95 |  * copy server message in the szBufer
96 |  */
97 | while((szBuf[ii++] = getchar()) != '\n');
98 |
99 | /**
100 |  * and send that szBufer to stClient
101 |  */
102 | write(iConnFd, szBuf, sizeof(szBuf));
103 |
104 | /**
105 |  * if msg contains "Exit" then server exit and chat ended.
106 |  */
107 | if(strncmp("exit", szBuf, 4) == 0)
108 | {
109 |     printf("Server Exit...\n");
110 |     break;
111 | }
112 | }
113 | }
114 |
115 | /*****
116 |  *
117 |  *                               main
118 |  *
119 |  *****/
120 | int main(int argc, char **argv)
121 | {
122 |     int iSockFd;
123 |     int iConnFd;
124 |     int iLength;
125 |     STRUCT_SOCKADDR_IN stServAdd;
126 |     STRUCT_SOCKADDR_IN stClient;
127 |
128 |     UNUSED(argc);
129 |     UNUSED(argv);
130 |
131 |     memset(&stServAdd, 0, sizeof(stServAdd));
132 |     memset(&stClient, 0, sizeof(stClient));
```

Exemplo em C - Servidor

```
133 |  
134 | /**  
135 |  * socket create and verification  
136 |  */  
137 | iSockFd = socket(AF_INET, SOCK_STREAM, 0);  
138 |  
139 | if(iSockFd == -1)  
140 | {  
141 |     fprintf(stderr, "E: socket creation failed...\n");  
142 |     exit(EXIT_FAILURE);  
143 | }  
144 | else  
145 | {  
146 |     printf("Socket successfully created..\n");  
147 | }  
148 |  
149 | bzero(&stServAdd, sizeof(stServAdd));  
150 |  
151 | /**  
152 |  * assign IP, PORT  
153 |  */  
154 | stServAdd.sin_family = AF_INET;  
155 | stServAdd.sin_addr.s_addr = htonl(INADDR_ANY);  
156 | stServAdd.sin_port = htons(PORT);  
157 |  
158 | /**  
159 |  * Binding newly created socket to given IP and verification  
160 |  */  
161 | if((bind(iSockFd, (PSTRUCT_SOCKADDR) &stServAdd, sizeof(stServAdd))) != 0)  
162 | {  
163 |     fprintf(stderr, "E: socket bind failed...\n");  
164 |  
165 |     exit(EXIT_FAILURE);  
166 | }  
167 |  
168 | printf("Socket successfully binded..\n");  
169 |  
170 | /**  
171 |  * Now server is ready to listen and verification  
172 |  */  
173 | if((listen(iSockFd, 5)) != 0)  
174 | {  
175 |     fprintf(stderr, "E: Listen failed...\n");  
176 |
```

Exemplo em C - Servidor

```
177 exit(EXIT_FAILURE);
178 }
179
180 printf("Server listening...\n");
181
182 iLength = sizeof(stClient);
183
184 /**
185  * Accept the data packet from stClient and verification
186  */
187 iConnFd = accept(iSockFd, (PSTRUCT_SOCKADDR) &stClient, (socklen_t *) &iLength);
188 if(iConnFd < 0)
189 {
190     fprintf(stderr, "E: server accept failed...\n");
191
192     exit(EXIT_FAILURE);
193 }
194
195 printf("server accept the stClient...\n");
196
197 /**
198  * Function for chatting between stClient and server
199  */
200 vChat(iConnFd);
201
202 /**
203  * After chatting close the socket
204  */
205 close(iSockFd);
206
207 return 0;
208 }
209
```

NORMAL server.c c utf-8[unix] 84% |i:177/209= %:1 [177]trailing [199]mixed-indent [178:180]mix-indent-file
"server.c" 209L, 5522B written

Exemplo em C - Cliente

```
1  /**
2   * client.c
3   *
4   * Developed by geek4geeks, modified by
5   * Gustavo Bacagine <gustavo.bacagine@protonmail.com>
6   *
7   * Example of server using sockets in C
8   *
9   * Compiling with: gcc -o server server.c -Wall -Wextra -O3 -ansi
10  *
11  * Date: 2023-10-31
12  */
13
14  /*****
15   *
16   *                               Includes
17   *
18   *****/
19  #include <stdio.h>
20  #include <stdbool.h>
21  #include <stdlib.h>
22  #include <string.h>
23  #include <strings.h> /* bzero() */
24  #include <arpa/inet.h> /* inet_addr() */
25  #include <netdb.h>
26  #include <sys/socket.h>
27  #include <unistd.h> /* read(), write(), close() */
28
29  /*****
30   *
31   *                               Defines and macros
32   *
33   *****/
34
35  /**
36   * Used to ignore warnings in compilation
37   * about unused variables
38   */
39  #define UNUSED(X) (void) X
40
41  /**
42   * Communication definitions
43   */
44  #define MAX          80
```

NORMAL client.c

c utf-8[unix] 0% |1:1/159= %:1 [4]trailing [138]mixed-indent [70:73]mix-indent-file

Exemplo em C - Cliente

```
45 #define IP    "127.0.0.1"
46 #define PORT    8080
47
48 /*****
49 *
50 *          Typedefs, structures, unions and enums
51 *
52 *****/
53 typedef struct sockaddr STRUCT_SOCKADDR;
54 typedef struct sockaddr * PSTRUCT_SOCKADDR;
55
56 typedef struct sockaddr_in STRUCT_SOCKADDR_IN;
57 typedef STRUCT_SOCKADDR_IN * PSTRUCT_SOCKADDR_IN;
58
59 /*****
60 *
61 *          Functions
62 *
63 *****/
64
65 /**
66 * Function designed for chat between stClientent and server.
67 */
68 void vChat(int iSockFd)
69 {
70     char szBuf[MAX];
71     int ii;
72
73     while(true)
74     {
75         bzero(szBuf, sizeof(szBuf));
76
77         printf("Type a string to send to the server: ");
78         ii = 0;
79
80         while((szBuf[ii++] = getchar()) != '\n');
81
82         write(iSockFd, szBuf, sizeof(szBuf));
83
84         bzero(szBuf, sizeof(szBuf));
85
86         read(iSockFd, szBuf, sizeof(szBuf));
87
88         printf("From Server : %s", szBuf);
```

Exemplo em C - Cliente

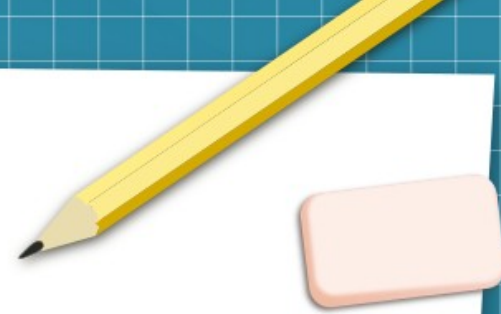
```
89
90     if((strcmp(szBuf, "exit", 4)) == 0)
91     {
92         printf("Client Exit...\n");
93         break;
94     }
95 }
96 }
97
98 /*****
99 *
100 *                               main
101 *
102 *****/
103 int main(int argc, char **argv)
104 {
105     int iSockFd;
106     STRUCT_SOCKADDR_IN stServAdd;
107     STRUCT_SOCKADDR_IN stClient;
108
109     UNUSED(argc);
110     UNUSED(argv);
111
112     memset(&stServAdd, 0, sizeof(stServAdd));
113     memset(&stClient, 0, sizeof(stClient));
114
115     /**
116      * socket create and verification
117      */
118     iSockFd = socket(AF_INET, SOCK_STREAM, 0);
119     if(iSockFd == -1)
120     {
121         fprintf(stderr, "E: socket creation failed...\n");
122         exit(EXIT_FAILURE);
123     }
124
125     printf("Socket successfully created...\n");
126
127     bzero(&stServAdd, sizeof(stServAdd));
128
129     /**
130      * assign IP, PORT
131      */
132     stServAdd.sin_family = AF_INET;
```

Exemplo em C - Cliente

```
133 stServAdd.sin_addr.s_addr = inet_addr(IP);
134 stServAdd.sin_port = htons(PORT);
135
136 /**
137  * connect the stClientent socket to server socket
138  */
139 if(connect(iSockFd, (PSTRUCT_SOCKADDR) &stServAdd, sizeof(stServAdd)) != 0)
140 {
141     fprintf(stderr, "E: connection with the server failed...\n");
142     exit(EXIT_FAILURE);
143 }
144
145 printf("connected to the server...\n");
146
147 /**
148  * vChattion for chat
149  */
150 vChat(iSockFd);
151
152 /**
153  * close the socket
154  */
155 close(iSockFd);
156
157 return 0;
158 }
159
```

NORMAL client.c c utf-8[unix] 83% |i:133/159= %:1 [144]trailing [138]mixed-indent [134:137]mix-indent-file
"client.c" 159L, 4501B written

Saída do exemplo em C



Porque Deus amou o mundo de tal maneira que deu o seu Filho unigênito, para que todo aquele que nele crê não pereça, mas tenha a vida eterna.

João 3:16

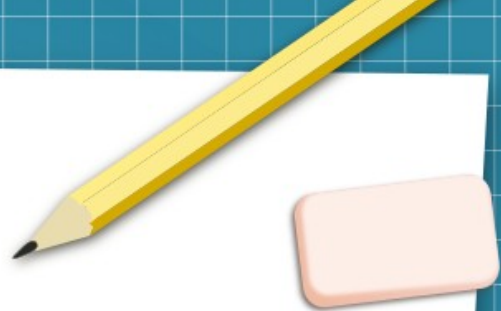
```
[bacagine@samsung-550xda]~  
$ ./server  
Socket successfully created..  
Socket successfully binded..  
Server listening..  
server accept the stClient...  
From Clientent: Ola servidor  
Type a string to send to the client: Ola cliente  
From Clientent: exit  
Type a string to send to the client: exit  
Server Exit...  
[bacagine@samsung-550xda]~  
$
```

Porque Deus amou o mundo de tal maneira que deu o seu Filho unigênito, para que todo aquele que nele crê não pereça, mas tenha a vida eterna.

João 3:16

```
[bacagine@samsung-550xda]~  
$ ./client  
Socket successfully created..  
connected to the server..  
Type a string to send to the server: Ola servidor  
From Server : Ola cliente  
Type a string to send to the server: exit  
From Server : exit  
Client Exit...  
[bacagine@samsung-550xda]~  
$
```


Fontes



- <https://www.geeksforgeeks.org/tcp-server-client-implementation-in-c/>
- Manpages do linux sobre as funções em C
- Slide “Modelo Cliente/Servidor em Sistemas Distribuídos” do professor Walter Matheos



This work is licensed under a Creative Commons
Attribution-ShareAlike 3.0 Unported License.
It makes use of the works of Mateus Machado Luna.

