

Handle multiple socket connections with fd_set and select on Linux

C

By [Silver Moon](#)

On Dec 25, 2011

14 Comments

Like 22

G+ 2

Tweet

Handle multiple socket connections

When writing server programs using sockets , it becomes necessary to handle multiple connections at a time , since a server needs to serve multiple clients.



There are many ways to do so. On linux this can be done in various ways like forking , threading , select method etc.

In this tutorial we shall use the select method approach. The **select** function allows the program to monitor multiple sockets for a certain "activity" to occur. For example if there is some data to be read on one of the sockets select will provide that information.

fd_set

An fd_set is a set of sockets to "monitor" for some activity. There are four useful macros : FD_CLR, FD_ISSET, FD_SET, FD_ZERO for dealing with an fd_set.

```
FD_ZERO - Clear an fd_set
FD_ISSET - Check if a descriptor is in an fd_set
FD_SET - Add a descriptor to an fd_set
FD_CLR - Remove a descriptor from an fd_set
```

```
1 //set of socket descriptors
2 fd_set readfds;
3
4 //socket to set
5 FD_SET( s , &readfds);
```

select function

The select method takes a list of socket for monitoring them. Here is how :

```
1 | activity = select( max_fd + 1 , &readfds , NULL , NULL , NULL);
```

The **select** function blocks , till an activity occurs. For example when a socket is ready to be read , select will return and readfds will have those sockets which are ready to be read.

Code

```
1 /**
2  Handle multiple socket connections with select and fd_set on Linux
3
4  Silver Moon ( m00n.silvr@gmail.com)
5 */
6
7 #include <stdio.h>
8 #include <string.h> //strlen
9 #include <stdlib.h>
10 #include <errno.h>
11 #include <unistd.h> //close
12 #include <arpa/inet.h> //close
13 #include <sys/types.h>
14 #include <sys/socket.h>
15 #include <netinet/in.h>
16 #include <sys/time.h> //FD_SET, FD_ISSET, FD_ZERO macros
17
```

Google™ Custom Search

SE



Download 10 Free
Linux Ebooks

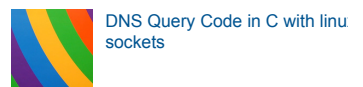
Connect with us



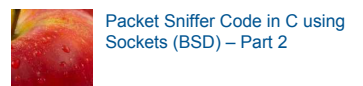
Other interesting stuff



[Code raw sockets in C on Linux](#)



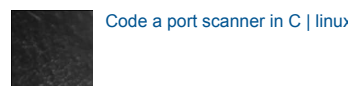
[DNS Query Code in C with linux sockets](#)



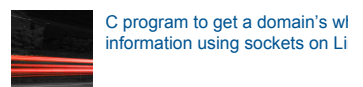
[Packet Sniffer Code in C using Sockets \(BSD\) – Part 2](#)



[SYN Flood DOS Attack with C Code \(Linux\)](#)



[Code a port scanner in C | linux](#)

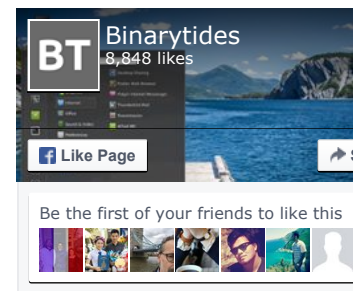


[C program to get a domain's w information using sockets on Li](#)

```

18 #define TRUE 1
19 #define FALSE 0
20 #define PORT 8888
21
22 int main(int argc , char *argv[])
23 {
24     int opt = TRUE;
25     int master_socket , addrlen , new_socket , client_socket[30] , max_clients = 30 ,
26     int max_sd;
27     struct sockaddr_in address;
28
29     char buffer[1025]; //data buffer of 1K
30
31     //set of socket descriptors
32     fd_set readfds;
33
34     //a message
35     char *message = "ECHO Daemon v1.0 \r\n";
36
37     //initialise all client_socket[] to 0 so not checked
38     for (i = 0; i < max_clients; i++)
39     {
40         client_socket[i] = 0;
41     }
42
43     //create a master socket
44     if( (master_socket = socket(AF_INET , SOCK_STREAM , 0)) == 0)
45     {
46         perror("socket failed");
47         exit(EXIT_FAILURE);
48     }
49
50     //set master socket to allow multiple connections , this is just a good habit, it
51     if( setsockopt(master_socket, SOL_SOCKET, SO_REUSEADDR, (char *)&opt, sizeof(opt))
52     {
53         perror("setsockopt");
54         exit(EXIT_FAILURE);
55     }
56
57     //type of socket created
58     address.sin_family = AF_INET;
59     address.sin_addr.s_addr = INADDR_ANY;
60     address.sin_port = htons( PORT );
61
62     //bind the socket to localhost port 8888
63     if (bind(master_socket, (struct sockaddr *)&address, sizeof(address))<0)
64     {
65         perror("bind failed");
66         exit(EXIT_FAILURE);
67     }
68     printf("Listener on port %d \n", PORT);
69
70     //try to specify maximum of 3 pending connections for the master socket
71     if (listen(master_socket, 3) < 0)
72     {
73         perror("listen");
74         exit(EXIT_FAILURE);
75     }
76
77     //accept the incoming connection
78     addrlen = sizeof(address);
79     puts("Waiting for connections ...");
80
81     while(TRUE)
82     {
83         //clear the socket set
84         FD_ZERO(&readfds);
85
86         //add master socket to set
87         FD_SET(master_socket, &readfds);
88         max_sd = master_socket;
89
90         //add child sockets to set
91         for ( i = 0 ; i < max_clients ; i++)
92         {
93             //socket descriptor
94             sd = client_socket[i];
95
96             //if valid socket descriptor then add to read list
97             if(sd > 0)
98                 FD_SET( sd , &readfds);
99
100             //highest file descriptor number, need it for the select function
101             if(sd > max_sd)
102                 max_sd = sd;
103         }
104
105         //wait for an activity on one of the sockets , timeout is NULL , so wait indef
106         activity = select( max_sd + 1 , &readfds , NULL , NULL , NULL);
107
108         if ((activity < 0) && (errno!=EINTR))
109         {
110             printf("select error");
111         }
112
113         //If something happened on the master socket , then its an incoming connectio
114         if (FD_ISSET(master_socket, &readfds))
115         {
116             if ((new_socket = accept(master_socket, (struct sockaddr *)&address, (soc
117             {
118                 perror("accept");
119                 exit(EXIT_FAILURE);
120             }
121

```



[Tokopedia.com](http://www.tokopedia.com/binarytides), tempat ngumpulnya online shop terpercaya se Indonesia dengan harga termurahESP! WiFi Module SMD tipe 12EBelanja o di Tokopedia.com? Lebih Lengkap, L Murah, Lebih Aman. Cek sekarang!C



```

122 //inform user of socket number - used in send and receive commands
123 printf("New connection , socket fd is %d , ip is : %s , port : %d \n" , n
124
125 //send new connection greeting message
126 if( send(new_socket, message, strlen(message), 0) != strlen(message) )
127 {
128     perror("send");
129 }
130
131 puts("Welcome message sent successfully");
132
133 //add new socket to array of sockets
134 for (i = 0; i < max_clients; i++)
135 {
136     //if position is empty
137     if( client_socket[i] == 0 )
138     {
139         client_socket[i] = new_socket;
140         printf("Adding to list of sockets as %d\n" , i);
141
142         break;
143     }
144 }
145 }
146
147 //else its some IO operation on some other socket :)
148 for (i = 0; i < max_clients; i++)
149 {
150     sd = client_socket[i];
151
152     if (FD_ISSET( sd , &readfds))
153     {
154         //Check if it was for closing , and also read the incoming message
155         if ((valread = read( sd , buffer, 1024)) == 0)
156         {
157             //Somebody disconnected , get his details and print
158             getpeername(sd , (struct sockaddr*)&address , (socklen_t*)&addrlen);
159             printf("Host disconnected , ip %s , port %d \n" , inet_ntoa(addr
160
161             //Close the socket and mark as 0 in list for reuse
162             close( sd );
163             client_socket[i] = 0;
164         }
165
166         //Echo back the message that came in
167         else
168         {
169             //set the string terminating NULL byte on the end of the data read
170             buffer[valread] = '\0';
171             send(sd , buffer , strlen(buffer) , 0 );
172         }
173     }
174 }
175 }
176
177 return 0;
178 }

```



The source code has been put up on the following url

<https://gist.github.com/silv3rm00n/5604330>

Compile and run the above program. Then connect to it using telnet from 3 different terminals.

```
$ telnet localhost 8888
```

Now whatever you type and send to server will be send back as it is, or echoed.

The server terminal would show details of connections like this :

```

Waiting for connections...
New connection , socket fd is 4 , ip is : 127.0.0.1 , port : 57831
Welcome message sent successfully
Adding to list of sockets as 0
New connection , socket fd is 5 , ip is : 127.0.0.1 , port : 57832
Welcome message sent successfully
Adding to list of sockets as 1
New connection , socket fd is 6 , ip is : 127.0.0.1 , port : 57833
Welcome message sent successfully
Adding to list of sockets as 2
New connection , socket fd is 7 , ip is : 127.0.0.1 , port : 57834
Welcome message sent successfully

```

The client terminal can be like this

```

$ telnet localhost 8888
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^'.

```

```
ECHO Daemon v1.0
ccc
ccc
ddd
ddd
fff
fff
```

There are other functions that can perform tasks similar to select. pselect , poll , ppoll

Resources

1. <http://pubs.opengroup.org/onlinepubs/7908799/xsh/select.html>
2. <http://linux.die.net/man/2/select>

Last Updated On : 18th May 2013

c linux socket programming

Subscribe to get updates delivered to your inbox

Enter email to subscribe

Subscribe

Related Posts

[10 examples of Linux ss command to monitor network connections](#)

[Socket programming in C on Linux – tutorial](#)

[Code a simple socket server in Python](#)

[PHP Socket programming tutorial](#)

[Perl socket programming tutorial](#)



About **Silver Moon**

Php developer, blogger and Linux enthusiast. He can be reached at admin@binarytides.com. Or find him on [Google+](#)



[Tokopedia.com, tempat ngumpulnya online shop terpercaya se-Indonesia dengan harga termurah](#)
MICRO USB TO LITHIUM CHARGING MODULE / MODUL CHARGER MICRO USB

14 Comments

BinaryTides

Login ▾

Recommend

Share

Sort by Best ▾



Join the discussion...



Jyx · 3 years ago

Nice article, but isn't it a problem in the part "//add new socket to array of sockets" where you assign max_clients? I.e. you set "i = max_clients;" what if a client with socket number less than max_clients has disconnected earlier. Then in the loop you will find it in the client_socket[i] which is equal to zero. Then you will add the new socket in that location and assign max_clients to that socket. However there could be sockets with higher id still available.

1 ^ | ▾ · Reply · Share ▸



Silver Moon Jyx · 3 years ago

i is just a temporary loop variable.

i = max_clients is used just to quit the loop, putting break instead of that would do the same thing.

and empty position in client_sockets array is searched everytime a new socket connection comes in and the loop starts i from 0.

^ | ▾ · Reply · Share ▸



VS · 4 months ago

I can multi clients connect but when I send data from client 1 only client 1 receive response from server. How to server response to all client connected?

^ | ▾ · Reply · Share ▸

**Stephan Berger** · a year ago

Thanx a lot for sharing!

Just one hint: If the client doesn't close the socket-connection properly, the prog will exit when sending to this connection (line 171). "Program received signal SIGPIPE, Broken pipe."
To avoid this You should use send() with the "MSG_NOSIGNAL" option:
send(sd , buffer , strlen(buffer) , MSG_NOSIGNAL);

Cheers, Stephan

[^](#) | [v](#) · [Reply](#) · [Share](#) ›**nmmm** · a year ago

it is bit unclear what will happen when client_socket array is full?
clients will be accepted using accept() but then they will appear nowhere?

[^](#) | [v](#) · [Reply](#) · [Share](#) ›**Chris** → **nmmm** · 8 months ago

I think because he didn't define a catch for that case, the program will automatically drop the new connection. Other examples of socket programming I've seen actually catch that case and allow the user to know gently what has happened.

<http://www.lowtek.com/sockets/...>[^](#) | [v](#) · [Reply](#) · [Share](#) ›**Luke** · a year ago

So, does the last for loop broadcast to all clients connected? How would you implement broadcasting to all but the one who sent the message?

[^](#) | [v](#) · [Reply](#) · [Share](#) ›**MhenryDpai** · 2 years ago

I get the success on connect message multiple times from different telnet sessions - however I do not get anything echoed back to telnet client

[^](#) | [v](#) · [Reply](#) · [Share](#) ›**Filip Jarno** · 2 years ago

The best explanation of sockets and select that I found, thanks!

[^](#) | [v](#) · [Reply](#) · [Share](#) ›**mohamad** · 3 years ago

i implemented such as this code, but when server send to client , client not recieved.why?!

[^](#) | [v](#) · [Reply](#) · [Share](#) ›**Silver Moon** Mod → **mohamad** · 3 years ago

is the client able to connect to server in the first place ?
does the server indicate that it received a new connection ?

if yes, then whatever message the client sends to server, it should get back the same and print it.

[^](#) | [v](#) · [Reply](#) · [Share](#) ›**Florian** · 3 years ago

Thanks!

You saved me MANY headaches.

[^](#) | [v](#) · [Reply](#) · [Share](#) ›**Sascha** · 3 years ago

wow, the article is perfect for me...

Thanks!

[^](#) | [v](#) · [Reply](#) · [Share](#) ›**Nishant** · 4 years ago

I go to this link while googling for "handling multiple socket connections". This is exactly what I was looking for. Thanks!

[^](#) | [v](#) · [Reply](#) · [Share](#) ›

Subscribe



Add Disqus to your site



Privacy

DISQUS

[About us](#) [Contact us](#) [Faq](#) [Advertise](#) [Privacy Policy](#)

Copyright © 2015 BinaryTides