21-09-2021

COMPUTER NETWORKS LAB - 3

OBSERVATION

SACHIN RAGHUL T

2019103573

Connecting Multiple clients with a single server using sockets.

Server.c

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <string.h>

#define PORT 3554
#define SIZE 256

int main()
{
    int serversocket, newsocket;
    struct sockaddr_in server_address, new_address;
    pid_t childpid;

    socklen_t serverLen = sizeof(server_address);
    socklen_t clientLen = sizeof(new_address);
    char msg[SIZE], buf[SIZE];
    serversocket = socket (AF_INET, SOCK_STREAM, 0);

    if (serversocket <0)
    {
        perror ("can't create socket... \n");
        return 1;
    }
    printf (" server socket created... \n");
    memset(&server_address, '\0', sizeof(server_address));

    server_address.sin_family = AF_INET;
    server_address.sin_port = htons(PORT);
    server_address.sin_addr.s.addr = INADDR_ANY;
```

```
int b = bind(serversocket, (struct sockaddr*)&server_address,
                  serverlen);

if(b<0)
{
   perror("Binding error...\n");
   return 1;
}
if (listen(serversocket, 2)==0)
{
   printf("Listening...\n");
}
else
{
   perror("cant listen to requests...\n");
   return 1;
}

while(1)
{
   newsocket = accept(serversocket, (struct sockaddr*)&new_address,
                  &clientLen);
   if(newsocket <0)
   {
      perror("cant accept requests...\n");
      return 1;
   }

   printf("connection accepted from %d:%d\n",
      inet_ntoa(new_address.sin_addr), ntohs(new_address.sinport));
   if((childpid =fork())==0)
   {
      close(serversocket);
      while(1)
      {
         recv(newSocket, msg, sizeof(msg), 0);
         if(strcmp(msg,"exit")==0)
         {
            printf("client %d diconnected from server...\n",
               inet_ntoa(new_address, sin_addr));
            return 1;
         }
         else
         {
            printf("client %d :%s\n",ntohs(new_address.sinport));
            sprintf(buf, "Server : %s\n", msg);
```

```
        send(newsocket, buf, strlen(buf), 0);
        bzero(buf, sizeof(buf));
      }
    }
  }
  return 0;
}
```

Client.c

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <string.h>

#define PORT 3554
#define SIZE 256

int main()
{
  int clientsocket;
  struct sockaddr_in server_address;
  char msg[SIZE], buf[SIZE];
  clientsocket = socket(AF_INET, SOCK_STREAM, 0);
   if(Clientsocket <0)
   {
     perror(" can't create socket...\n");
     return 1;
   }
   printf(" client socket created...\n");
   memset(& server_address, '\0', sizeof(server_address));
   server_address.sin_family = AF_INET;
   Serrver_address.sin_port = htons(PORT);
   serrver_address.sin_addr.s_addr = INADDR_ANY;
   int c = connect(clientsocket,(struct sockaddr*)&server_address
               , sizeof(server_address));
if(c<0)
{
    perror("cant connect to the server...\n");
    return 1;
}
printf("connected to the server..\n");
while(1)
{
    printf("Enter your message and press enter...\n");
    scanf("%s", buf);
    send(clientsocket, buf, sizeof(buf),0);
    if(strcmp(buf,"exit")==0)
    {
        close(clientsocket);
        printf("Disconnected from the server...\n");
        return 1;
    }
    if(recv(clientsocket, msg, sizeof(msg),0)<0)
    {
        perror("can't receive message...\n");
        return 1;
    }
    printf("%s\n", msg);
    bzero(msg, sizeof(msg));
}
return 0;
}
```

OUTPUT :

client 1:

```
Client socket created
connected to the server
Enter your message...
sachin
server : sachin
Enter your message...
Hey
server : Hey
Enter your message...
exit
Disconnected from the server...
```

Client 2 :

```
client socket created
Connected to the server
Enter your message...
Rohit
Server : Rohit
Enter your message...
Hii
server : Hii
Enter your message...
exit
Disconnected from the server
```

client 3:

```
client socket created
connected to the server
Enter your message...
virat
server: virat
Enter your message...
Hello
server: Hello
Enter your message...
exit
Disconnected from the server
```

Server:

```
Server socket created...
Listening...
connection accepted from : 42182
connection accepted from : 42204
connection accepted from : 42248
client 42182 : sachin
client 42182 : Hey
client 42204 : Rohit
client 42204 : Hii
client 42248 : virat
client 42248 : Hello
client disconnected from the server...
client disconnected from the server...
client disconnected from the server...
```