CSE1004 NAC LAB
ASSIGNMENT NO. – 6

UDP SOCKET PROGRAMMING

DATE – 10/02/22

NAME – AYUSHI TRIVEDI

REGISTRATION NUMBER – 20BPS1135

Develop and execute a UDP socket program to compute the sum of first 'n' even numbers. The server calculates the sum based on the 'n' value received from the client and the computed sum value should be displayed at the client side.

SERVER:

ALGORITHM:

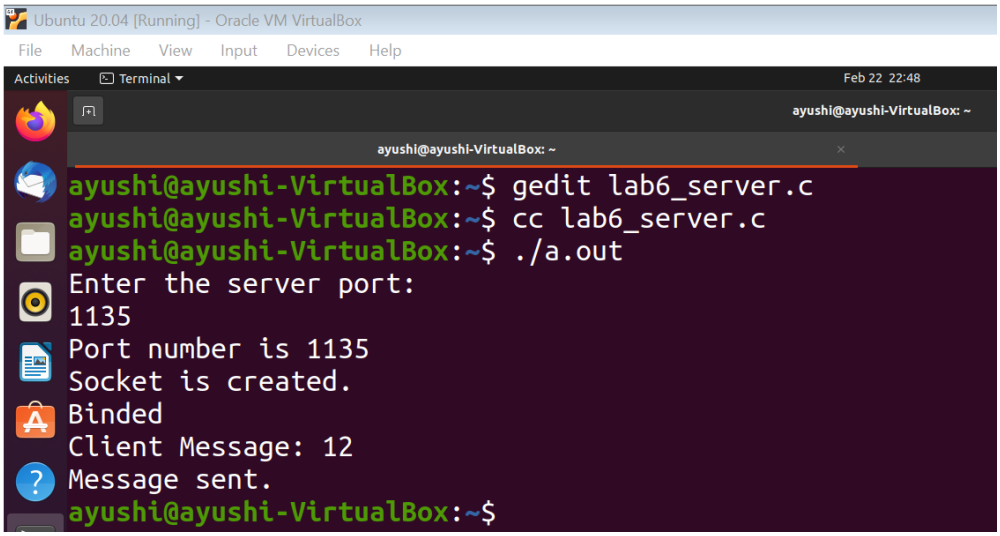
1. Defining MAXLINE globally.
2. Creating two char variables rev_msg[MAXLINE] and send_msg[MAXLINE], one for receiving data from client and one for sending.
3. Reading server **port** number.
4. Creating a socket using **socket()** function and filling server information
5. Binding the socket to the server address using the **bind()** function.
6. Receiving input number from client using **recvfrom()** function and storing it in rec_msg.
7. Calculating sum of first input number and storing it in send_msg
8. Sending send_msg to client using **sendto()** function.
9. Closing server socket using **close()** function.

CODE:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netinet/in.h>
#define MAXLINE 1024
// Driver code
int main() {
   int sockfd,PORT,sd2;
   char rev_msg[MAXLINE];
   char sent_msg[MAXLINE]="xd";
   struct sockaddr_in servaddr, cliaddr;
   printf("Enter the server port: \n");
         scanf("%d",&PORT);
         printf("Port number is %d\n",PORT);
   sockfd = socket(AF_INET, SOCK_DGRAM, 0);
 // Creating socket file descriptor
   if(sockfd<0) printf("Can't create. \n");
   else printf("Socket is created. \n");
   memset(&servaddr, 0, sizeof(servaddr));
   memset(&cliaddr, 0, sizeof(cliaddr));
 // Filling server information
   servaddr.sin_family = AF_INET; // IPv4
   servaddr.sin_addr.s_addr = INADDR_ANY;
   servaddr.sin_port = htons(PORT);
```

```
// Bind the socket with the server address
   sd2=bind(sockfd, (const struct sockaddr *)&servaddr,sizeof(servaddr));
   if(sd2<0) printf("Can't Bind.\n");
   else{
         printf("Binded\n");
         int len, n;
      len = sizeof(cliaddr); //len is value/resuslt
      n = recvfrom(sockfd, (char *)rev_msg, MAXLINE,
      MSG_WAITALL, ( struct sockaddr *) &cliaddr,&len);
      rev_msg[n] = '\0';
      printf("Client Message: %s\n", rev_msg);
      int num=atoi(rev_msg);
      int sum= (num*(num+1));
      sprintf(sent_msg,"%d",sum);
      sendto(sockfd, (const char *)sent_msg, strlen(sent_msg),MSG_CONFIRM, (const struct
sockaddr *) &cliaddr,len);
      printf("Message sent.\n");
   }
         return 0;
}
```

OUTPUT:



CLIENT:

ALGORITHM:

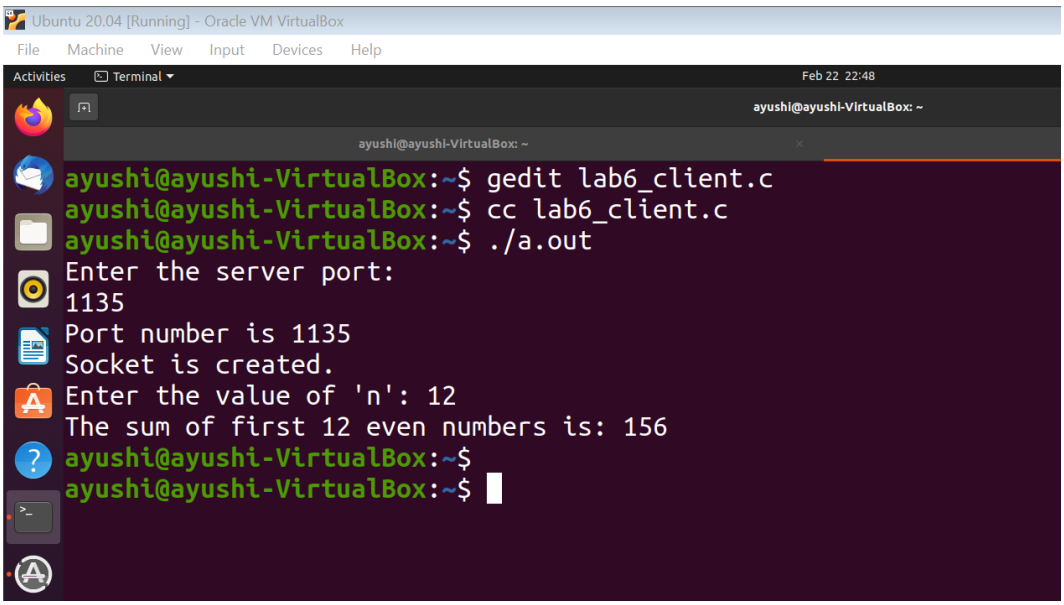
1. Reading client **port** number.
2. Creating two char variables rev_msg[MAXLINE] and send_msg[MAXLINE], one for receiving data from server and one for sending.
3. Creating socket using **socket()** function.
4. Filling server information.
5. Asking user to enter the value of n and storing it in send_msg.
6. Sending send_msg to server using **sendto()** function.
7. Receiving result from server using **recvfrom()** function and storing it in rec_msg.
8. Printing the result.
9. Closing client socket using **close()** function.

CODE:

```
#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#include <netinet/in.h>


// Driver code

int main() {

   int sockfd,PORT;

   char rev_msg[1024];

   char send_msg[1024];

   struct sockaddr_in  servaddr;
```

```
   printf("Enter the server port: \n");
          scanf("%d",&PORT);
          printf("Port number is %d\n",PORT);
          sockfd = socket(AF_INET, SOCK_DGRAM, 0);
          if(sockfd<0) printf("Can't create.\n");
          else{
                printf("Socket is created.\n");
                memset(&servaddr, 0, sizeof(servaddr));

 // Filling server information
       servaddr.sin_family = AF_INET;
       servaddr.sin_port = htons(PORT);
       servaddr.sin_addr.s_addr = INADDR_ANY;
       int n, len;
       printf("Enter the value of 'n': ");
       scanf("%s",send_msg);
       sendto(sockfd, (const char *)send_msg, strlen(send_msg),
       MSG_CONFIRM, (const struct sockaddr *) &servaddr,sizeof(servaddr));
       n = recvfrom(sockfd, (char *)rev_msg, 1024,MSG_WAITALL, (struct sockaddr *)
&servaddr,&len);
       rev_msg[n] = '\0';
       printf("The sum of first %s even numbers is: %s\n",send_msg, rev_msg);
       close(sockfd);
   }
   return 0;
}
```

OUTPUT:


Ubuntu 20.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Feb 22 22:48
ayushi@ayushi-VIrtualBox: ~
ayushi@ayushi-VIrtualBox: ~
ayushi@ayushi-VirtualBox:~$ gedit lab6_client.c
ayushi@ayushi-VirtualBox:~$ cc lab6_client.c
ayushi@ayushi-VirtualBox:~$ ./a.out
Enter the server port:
1135
Port number is 1135
Socket is created.
Enter the value of 'n': 12
The sum of first 12 even numbers is: 156
ayushi@ayushi-VirtualBox:~$
ayushi@ayushi-VirtualBox:~$