Name: VENNELA G

Register No: 20BDS0146

Lab Course Name: Network & Communication

Lab Slot: L20+L21

Assessment Title: Socket Programming

- socket() –Endpoint for communication

  Syntax: int sockid = socket(domain, type, protocol);

- bind() – Assign a unique telephone number

  Syntax: int status = bind(sockid, &addrport, size);

- listen()- Wait for a caller

  Syntax: int status = listen(sockid , queuelen);

- connect()-Dial a number

  Syntax: int status = connect(sockid, &addr, addrlen);

- accept()-Receive a call
  Syntax: int s = accept(sockid, &addr, &addrlen);

- send(), recv() – Talk

  Syntax: int count = send(sockid, &buf, len, flags);

  Syntax: int count = recv(sockid, &buf, len, flags);

- close()-Hang up

  Syntax: status = close(s);

20BDS0146
VENNELA G

Socket Programming in C

PROGRAM:

server.c (Server program)

```
/****************** SERVER CODE ****************/

#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <string.h>
#include<arpa/inet.h>
int main(){
  int welcomeSocket, newSocket;
  char buffer[1024];
  struct sockaddr_in serverAddr;
  struct sockaddr_storage serverStorage;
  socklen_t addr_size;

  /*---- Create the socket. The three arguments are: ----*/
  /* 1) Internet domain 2) Stream socket 3) Default protocol (TCP in
this case) */
  welcomeSocket = socket(PF_INET, SOCK_STREAM, 0);
/*---- Configure settings of the server address struct ----*/
```

```
  /* Address family = Internet */
  serverAddr.sin_family = AF_INET;
  /* Set port number, using htons function to use proper byte order
*/
  serverAddr.sin_port = htons(7891);
  /* Set IP address to localhost */
  serverAddr.sin_addr.s_addr = inet_addr("127.0.0.1");
  /* Set all bits of the padding field to 0 */
  memset(serverAddr.sin_zero, '\0', sizeof serverAddr.sin_zero);

  /*---- Bind the address struct to the socket ----*/
  bind(welcomeSocket, (struct sockaddr *) &serverAddr,
sizeof(serverAddr));

  /*---- Listen on the socket, with 5 max connection requests queued
----*/
  if(listen(welcomeSocket,5)==0)
    printf("Listening\n");
  else
  printf("Error\n");

  /*---- Accept call creates a new socket for the incoming connection -
---*/
  addr_size = sizeof serverStorage;
```

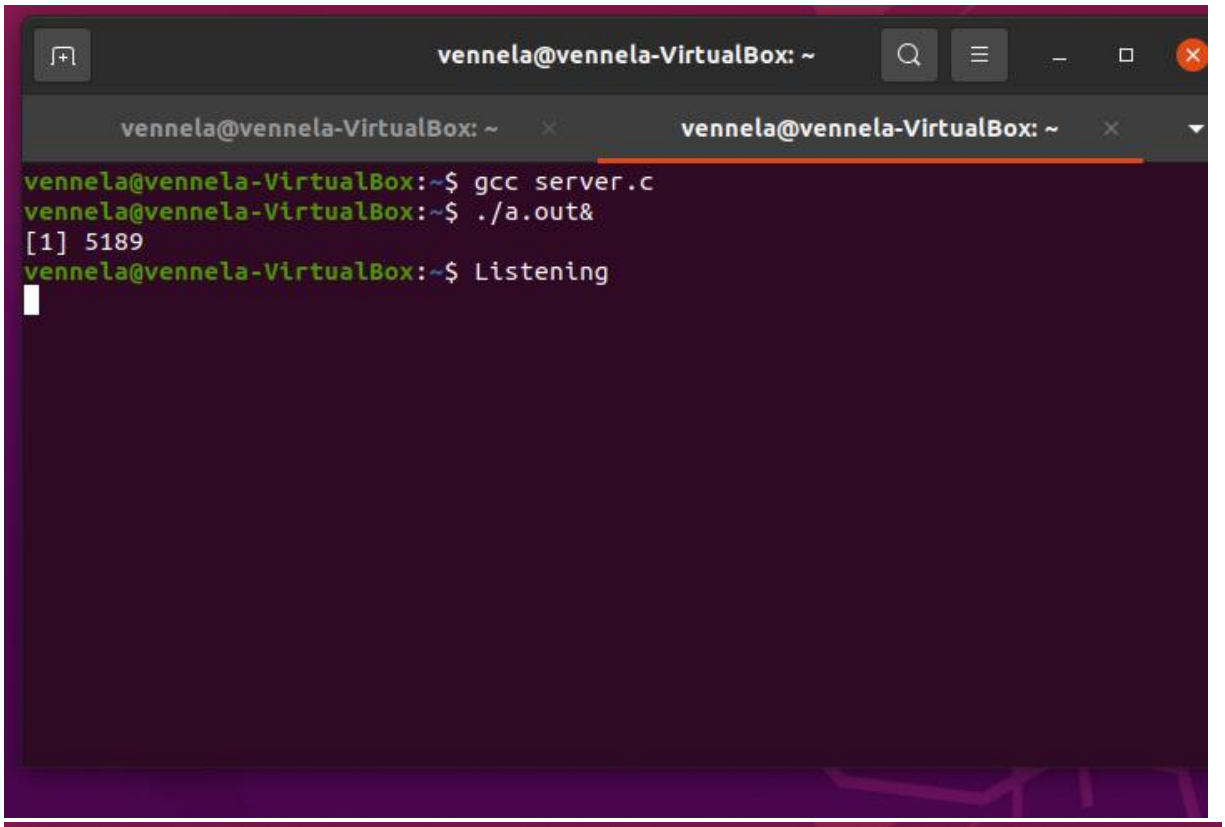
```
  newSocket = accept(welcomeSocket, (struct sockaddr *)
&serverStorage, &addr_size);


  /*---- Send message to the socket of the incoming connection ----*/
  strcpy(buffer,"Hello,This is 20BDS0146 from VIT\n");
  send(newSocket,buffer,33,0);


  return 0;
}
```

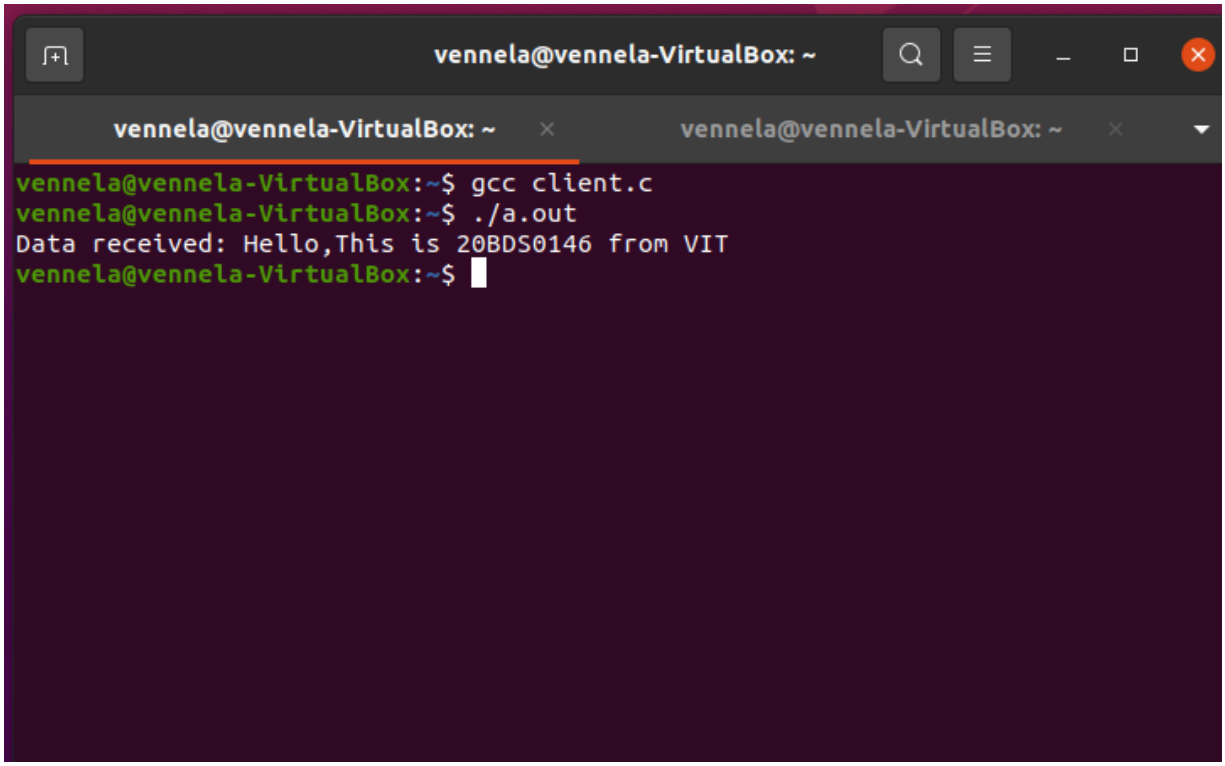
client.c (Client program)

```
/****************** CLIENT CODE ****************/

#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <string.h>
#include<arpa/inet.h>
int main(){
  int clientSocket;
  char buffer[10249];
  struct sockaddr_in serverAddr;
  socklen_t addr_size;

  /*---- Create the socket. The three arguments are: ----*/
  /* 1) Internet domain 2) Stream socket 3) Default protocol (TCP in
this case) */
  clientSocket = socket(PF_INET, SOCK_STREAM, 0);

  /*---- Configure settings of the server address struct ----*/
  /* Address family = Internet */
  serverAddr.sin_family = AF_INET;
/* Set port number, using htons function to use proper byte order */
```

```
  serverAddr.sin_port = htons(7891);
  /* Set IP address to localhost */
  serverAddr.sin_addr.s_addr = inet_addr("127.0.0.1");
  /* Set all bits of the padding field to 0 */
  memset(serverAddr.sin_zero, '\0', sizeof serverAddr.sin_zero);

  /*---- Connect the socket to the server using the address struct ----
*/
  addr_size = sizeof serverAddr;
  connect(clientSocket, (struct sockaddr *) &serverAddr, addr_size);

  /*---- Read the message from the server into the buffer ----*/
  recv(clientSocket, buffer, 10249, 0);

  /*---- Print the received message ----*/
  printf("Data received: %s",buffer);
return 0;
}
```

OUTPUT:


vennela@vennela-VirtualBox: ~
vennela@vennela-VirtualBox: ~
vennela@vennela-VirtualBox: ~
vennela@vennela-VirtualBox:~$ gcc server.c
vennela@vennela-VirtualBox:~$ ./a.out&
[1] 5189
vennela@vennela-VirtualBox:~$ Listening


vennela@vennela-VirtualBox: ~
vennela@vennela-VirtualBox: ~
vennela@vennela-VirtualBox: ~
vennela@vennela-VirtualBox:~$ gcc client.c
vennela@vennela-VirtualBox:~$ ./a.out
Data received: Hello,This is 20BDS0146 from VIT
vennela@vennela-VirtualBox:~$

20BDS0146
VENNELA G