DHRUV GARG   16 BCE 1190

NETWORK AND COMMUNICATION
LAB 5

IMPLEMENTATION OF STOP AND WAIT USING ARQ

SERVER SIDE CODE

```
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <time.h>
int main()
{
   time_t send_time,current_time;
   int PORT=1809,count=0,n,flag;
   char msg[1024];
   char msg1[1];
   struct sockaddr_in addr;
   int s = socket(AF_INET, SOCK_STREAM, 0);
   if(s<0)
   {
      printf(" Socket not created\n");
   }
   else
   {
      printf("Socket created\n");
   }

   addr.sin_port = htons(PORT);
   addr.sin_family = AF_INET;
   addr.sin_addr.s_addr = htonl(INADDR_ANY);

   int bind_var = bind(s, (struct sockaddr *)&addr,sizeof(struct sockaddr_in) );
   if(bind_var <0)
   {
      printf("Error binding socket\n");
   }
   else
   {
      printf("Successfully bound to port %u\n", PORT);
   }

   int ls = listen(s,5);
   if(ls<0)
   {
```

```
   printf("Listening Failed\n");
}
else
{
   printf("Listening to socket \n");
}

int length = sizeof(struct sockaddr_in);
int accept_var = accept(s, (struct sockaddr *)&addr,&length);
if(accept_var == -1)
{
   printf("Accept Failed\n");
}
else
{
   printf("Accepted the socket \n");
}
printf("Client Address = %s\n\n",inet_ntoa(addr.sin_addr));
printf("Enter no. of frames you want to sent\n");
scanf("%d",&n);
while(n!=0)
{

   printf("Enter Data for frame %d\n",count);
   scanf("%s",msg);
   flag=0;
   while(flag!=1)
   {
      send(accept_var, msg, 1024, 0);
      send_time=time(NULL);
      current_time=time(NULL);
      while(send_time-current_time<7200)
      {
         if(recv(accept_var, msg1, 1, 0))
         {
            break;
         }
         current_time=time(NULL);
      }
      if(count==0&&msg1[0]=='1')
      {
         flag=1;
         count=1;
      }
      if(count==1&&msg1[0]=='0')
      {
         flag=1;
         count=0;
         }
      }
      n--;
   }
}
```

CLIENT SIDE CODE

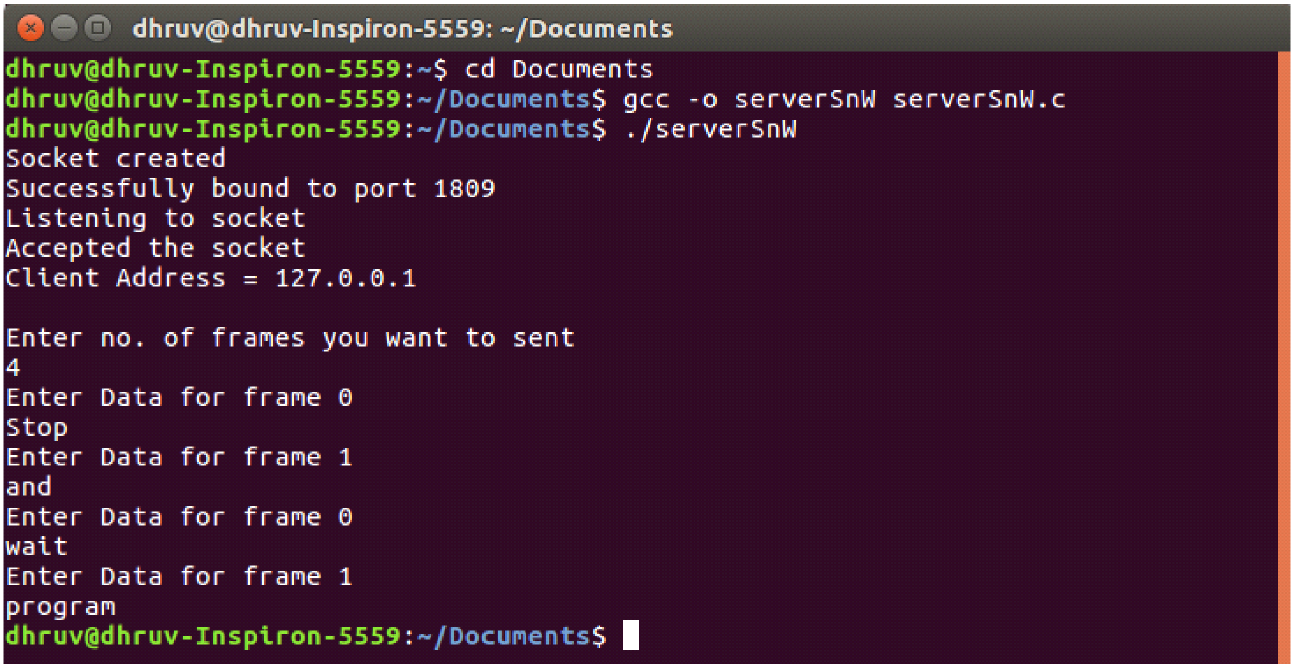
```
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <netinet/in.h>
int main()
{
   int PORT=1809;
   int count=0;
   char msg[1024],msg_old[1024],msg1[1];
   strcpy(msg,"");
   strcpy(msg_old,"");
   struct sockaddr_in addr;
   int s = socket(AF_INET, SOCK_STREAM, 0);
   if(s<0)
   {
      printf("Error, Port not created\n");
   }
   else
   {
      printf("Port created\n");
   }

   addr.sin_port = htons(PORT);
   addr.sin_family = AF_INET;
   addr.sin_addr.s_addr = htons(INADDR_ANY);

   int con=connect(s, (struct sockaddr *) &addr, sizeof(struct sockaddr_in));
   if(con<0)
   {
      printf("Connection failed\n");
   }
   else
   {
      printf("Connected to port %d\n\n",PORT);
   }
```

```
    while(1)
    {
        if(recv(s, msg, 1024, 0))
        {
            if(strcmp(msg,msg_old))
            {
                send(s, msg1, 1, 0);
                strcpy(msg_old,msg);
            }
            else
            {
                printf("Server : %s\n",msg);
                printf("Enter next frame either 0 or 1 : ");
                scanf(" %c",msg1);
                send(s, msg1, 1, 0);
                strcpy(msg_old,msg);
            }
        }
    }
}
```

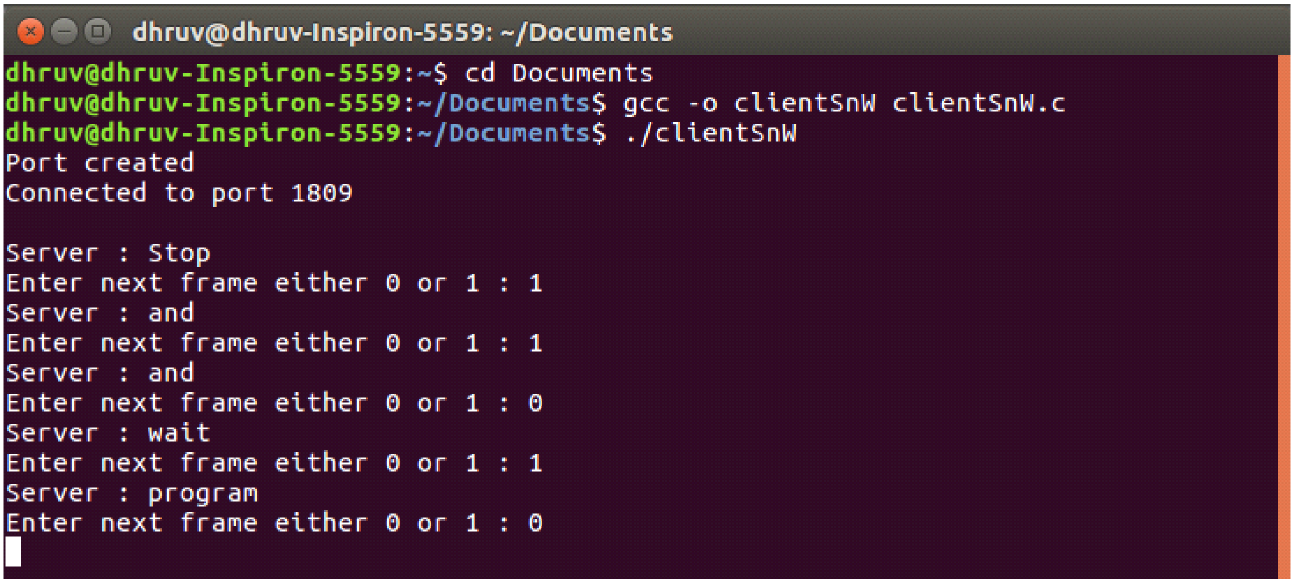
SCREENSHOT SERVER SIDE



```
dhruv@dhruv-Inspiron-5559:~$ cd Documents
dhruv@dhruv-Inspiron-5559:~/Documents$ gcc -o serverSnW serverSnW.c
dhruv@dhruv-Inspiron-5559:~/Documents$ ./serverSnW
Socket created
Successfully bound to port 1809
Listening to socket
Accepted the socket
Client Address = 127.0.0.1

Enter no. of frames you want to sent
4
Enter Data for frame 0
Stop
Enter Data for frame 1
and
Enter Data for frame 0
wait
Enter Data for frame 1
program
dhruv@dhruv-Inspiron-5559:~/Documents$
```

SCREENSHOT CLIENT SIDE



```
dhruv@dhruv-Inspiron-5559:~$ cd Documents
dhruv@dhruv-Inspiron-5559:~/Documents$ gcc -o clientSnW clientSnW.c
dhruv@dhruv-Inspiron-5559:~/Documents$ ./clientSnW
Port created
Connected to port 1809

Server : Stop
Enter next frame either 0 or 1 : 1
Server : and
Enter next frame either 0 or 1 : 1
Server : and
Enter next frame either 0 or 1 : 0
Server : wait
Enter next frame either 0 or 1 : 1
Server : program
Enter next frame either 0 or 1 : 0
```