Network Programs

TCP SERVER

```
#include<stdio.h>
#include<stdlib.h>
#include<netdb.h>
#include<sys/socket.h>
#include<string.h>

int main(){
  //variable declaration
  int serversocket,clientsocket,port;
  struct sockaddr_in serveraddr,clientaddr;
  socklen_t len = sizeof(clientaddr);
  char message[50];

  // socket creation & port number
  serversocket = socket(AF_INET,SOCK_STREAM,0);
  printf("Enter the port number : ");
  scanf("%d",&port);

  //server configuration
  serveraddr.sin_family = AF_INET;
  serveraddr.sin_port = htons(port);
  serveraddr.sin_addr.s_addr = INADDR_ANY;

  //binding socket
  bind(serversocket,(struct sockaddr*)&serveraddr,sizeof(serveraddr));

  //listen and accept client
  listen(serversocket,5);
  printf("Waiting for client...\n");
  clientsocket = accept(serversocket,(struct sockaddr*)&clientaddr,&len);
  printf("client connected\n");

  //read from client
  read(clientsocket,message,sizeof(message));
  printf("Client sent : %s\n",message);

  //write to client
printf("Sending acknowledgement to client\n");
write(clientsocket,"message_received",sizeof("message_received"));

  //close
  close(clientsocket);
  close(serversocket);
}
```

TCP CLIENT

```
#include<stdio.h>
#include<netinet/in.h>
#include<netdb.h>
#include<strings.h>
#include<sys/types.h>

int main(){

  //variable declaration
  int clientsocket,port;
  struct sockaddr_in serveraddr;
  char message[50];

  //socket and port
  clientsocket = socket(AF_INET,SOCK_STREAM,0);
  printf("Enter port : ");
  scanf("%d",&port);

  //server configuration
  serveraddr.sin_family = AF_INET;
  serveraddr.sin_port = htons(port);

  //connect to server
  connect(clientsocket,(struct sockaddr*)&serveraddr,sizeof(serveraddr));
  printf("Connected to server.\n");

  //send to server
  send(clientsocket,"Im client",sizeof("im client"),0);
  printf("Message sent\n");

  //recv ack from server
  recv(clientsocket,message,sizeof(message),0);
  printf("From server : %s\n",message);
  close(clientsocket);

}
```

TCP ADDITION SERVER

```
#include<stdio.h>
#include<stdlib.h>
#include<netdb.h>
#include<sys/socket.h>
#include<string.h>
```

```
int main(){
  int serversocket,clientsocket,port;
  struct sockaddr_in serveraddr,clientaddr;
  socklen_t len = sizeof(clientaddr);
  int a,b,sum = 0;

  serversocket = socket(AF_INET,SOCK_STREAM,0);
  printf("Enter port number : ");
  scanf("%d",&port);

  serveraddr.sin_family = AF_INET;
  serveraddr.sin_port = htons(port);
  serveraddr.sin_addr.s_addr = INADDR_ANY;

  bind(serversocket,(struct sockaddr*)&serveraddr,sizeof(serveraddr));
  listen(serversocket,5);
  printf("Waiting for client\n");

  clientsocket = accept(serversocket,(struct sockaddr*)&clientaddr,&len);
  printf("Client connected\n");

  read(clientsocket,&a,sizeof(&a));
  read(clientsocket,&b,sizeof(&b));
  printf("Client sent %d and %d\n",a,b);

  sum = a+b;
  write(clientsocket,&sum,sizeof(&sum));
  printf("Sum calculated and sent to client\n");
  close(clientsocket);
  close(serversocket);

}
```

TCP ADDITION CLIENT

```
#include<stdio.h>
#include<netinet/in.h>
#include<netdb.h>
#include<strings.h>
#include<sys/types.h>

int main(){
  int clientsocket,port;
  struct sockaddr_in serveraddr;
  int a,b,sum = 0;

  clientsocket = socket(AF_INET,SOCK_STREAM,0);
  printf("Enter port number : ");
  scanf("%d",&port);

  serveraddr.sin_family = AF_INET;
  serveraddr.sin_port = htons(port);

  connect(clientsocket,(struct sockaddr*)&serveraddr,sizeof(serveraddr));
  printf("Connected to the server\n");
  printf("Enter two numbers : ");
  scanf("%d%d",&a,&b);

  send(clientsocket,&a,sizeof(&a),0);
  send(clientsocket,&b,sizeof(&b),0);
  printf("Numbers send to server\n");

  recv(clientsocket,&sum,sizeof(&sum),0);
  printf("Sum received from server : %d\n",sum);

  close(clientsocket);
}
```

UDP SERVER

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

int main(){
  //variable declaration
  int serversocket,port;
  struct sockaddr_in serveraddr,clientaddr;
  socklen_t len = sizeof(clientaddr);
  char message[50];

  //socket & port
  serversocket = socket(AF_INET,SOCK_DGRAM,0);
  printf("Enter port number : ");
  scanf("%d",&port);

  //server conifg
  serveraddr.sin_family = AF_INET;
  serveraddr.sin_port = htons(port);
  serveraddr.sin_addr.s_addr = INADDR_ANY;

  //binding
  bind(serversocket,(struct sockaddr*)&serveraddr,sizeof(serveraddr));
  printf("Waiting for client connection\n");

  //recv from client
  recvfrom(serversocket,message,sizeof(message),0,(struct sockaddr*)&clientaddr,&len);
  printf("Connected to client and received message : %s\n",message);

  //sent to client
  sendto(serversocket,"message_received",sizeof("message_received"),0,(struct
sockaddr*)&clientaddr,len);
  printf("Acknowledgement sent to client\n");

  close(serversocket);

}
```

UDP CLIENT

```
#include<netinet/in.h>
#include<netdb.h>
#include<stdio.h>
#include<string.h>

int main(){
  //variables
  int clientsocket,port;
  struct sockaddr_in serveraddr;
  socklen_t len = sizeof(serveraddr);
  char message[50];

  //socket & port
  clientsocket = socket(AF_INET,SOCK_DGRAM,0);
  printf("Enter port : ");
  scanf("%d",&port);

  //server conifg
  serveraddr.sin_family = AF_INET;
  serveraddr.sin_port = htons(port);
  serveraddr.sin_addr.s_addr = INADDR_ANY;

  //send to server
  sendto(clientsocket,"Hi,im client",sizeof("Hi,im client"),0,(struct
sockaddr*)&serveraddr,sizeof(serveraddr));
  printf("Message sent to server\n");

  //recv from server
  recvfrom(clientsocket,message,sizeof(message),0,(struct sockaddr*)&serveraddr,&len);
  printf("Message from server : %s\n",message);
  close(clientsocket);
}
```

fact-udp SERVER

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

int main(){
  //variable
  int serversocket,port;
  struct sockaddr_in serveraddr,clientaddr;
  socklen_t len = sizeof(clientaddr);
  int a,fact = 1,i;

  //socket & port
  serversocket = socket(AF_INET,SOCK_DGRAM,0);
  printf("Enter port number : ");
  scanf("%d",&port);

  //config server
  serveraddr.sin_family = AF_INET;
  serveraddr.sin_port = htons(port);
  serveraddr.sin_addr.s_addr = INADDR_ANY;

  //bind
  bind(serversocket,(struct sockaddr*)&serveraddr,sizeof(serveraddr));
  printf("Waiting for client\n");

  //recv from client
  recvfrom(serversocket,&a,sizeof(a),0,(struct sockaddr*)&clientaddr,&len);
  printf("Received %d from client\n",a);

  //logic for factorial
  for(i = 1;i<=a;i++){
    fact*=i;
  }

  //send to client
  sendto(serversocket,&fact,sizeof(fact),0,(struct sockaddr*)&clientaddr,len);
  printf("Factorial Calculated and sent to client\n");

  close(serversocket);
}
```

FACT-UDP-CLIENT:

```
#include<stdio.h>
#include<netinet/in.h>
#include<netdb.h>
#include<unistd.h>
#include<string.h>

int main(){
  //variables
  int clientsocket,port;
  struct sockaddr_in serveraddr;
  socklen_t len = sizeof(serveraddr);
  int a,fact;

  //socket & port
  clientsocket = socket(AF_INET,SOCK_DGRAM,0);
  printf("Enter port : ");
  scanf("%d",&port);

  //config server
  serveraddr.sin_family = AF_INET;
  serveraddr.sin_port = htons(port);
  serveraddr.sin_addr.s_addr = INADDR_ANY;

  //read & send to server
  printf("Enter the number : ");
  scanf("%d",&a);
  sendto(clientsocket,&a,sizeof(a),0,(struct sockaddr*)&serveraddr,sizeof(serveraddr));
  printf("%d sent to server for calculation\n",a);

  //recv from server
  recvfrom(clientsocket,&fact,sizeof(fact),0,(struct sockaddr*)&serveraddr,&len);
  printf("Factorial received from server : %d\n",fact);

  close(clientsocket);
}
```

STOP & WAIT - SERVER

```
//Necessary libraries
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
```

```
#include<sys/socket.h>
#include<unistd.h>
#include<arpa/inet.h>

//packet structure
typedef struct packet{
    char data[1024];
}Packet;

//frame structure
typedef struct frame{
    int frame_kind;
    int sq_no;
    int ack;
    Packet packet;
}Frame;

int main(){
    //variable declaration
    int serversocket,port;
    struct sockaddr_in serveraddr,clientaddr;
    socklen_t len = sizeof(clientaddr);
    int frameid = 0;
    Frame frame_recv;
    Frame frame_send;

    //socket creation & port
    serversocket = socket(AF_INET,SOCK_DGRAM,0);
    printf("Enter the port no : ");
    scanf("%d",&port);

    //server configuration
    serveraddr.sin_family = AF_INET;
    serveraddr.sin_port = htons(port);
    serveraddr.sin_addr.s_addr = INADDR_ANY;

    //binding socket
    bind(serversocket,(struct sockaddr*)&serveraddr,sizeof(serveraddr));

    //logic of serverside stop&wait
    while(1){

             //recv just like udp prograM
        int frecv_size = recvfrom(serversocket,&frame_recv,sizeof(frame_recv),0,(struct
sockaddr*)&clientaddr,&len);

        if(frecv_size > 0 && frame_recv.frame_kind == 1 && frame_recv.sq_no == frameid){
            printf("Frame received: %s\n",frame_recv.packet.data);
            frame_send.sq_no = 0;
            frame_send.frame_kind = 0;
            frame_send.ack = frame_recv.sq_no+1;

            //send just like udp program
            sendto(serversocket,&frame_send,sizeof(frame_send),0,(struct
sockaddr*)&clientaddr,len);

            printf("Ack send\n");
        }
        else{
            printf("Frame not received\n");
        }
        frameid++;
    }
    //close
    close(serversocket);
}
```

STOP & WAIT - CLIENT

```
//Necessary headers
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<unistd.h>
#include<arpa/inet.h>
#include<sys/socket.h>

//packet
typedef struct packet{
    char data[1024];
}Packet;

//frame
typedef struct frame{
    int frame_kind;
    int sq_no;
    int ack;
    Packet packet;
}Frame;

int main(){
    //variable declaration
    int clientsocket,port;
    struct sockaddr_in serveraddr;
    socklen_t len = sizeof(serveraddr);
    char buffer[1024];
    int frameid = 0;
    Frame frame_send;
    Frame frame_recv;
    int ack_recv = 1;

    //socket & port
    clientsocket = socket(AF_INET,SOCK_DGRAM,0);
    printf("Enter port : ");
    scanf("%d",&port);

    //server config
    serveraddr.sin_family = AF_INET;
    serveraddr.sin_port = htons(port);
    serveraddr.sin_addr.s_addr = INADDR_ANY;

    //main logic of client side - stop & wait
    while(1){
        if(ack_recv == 1){
            frame_send.sq_no = frameid;
            frame_send.frame_kind = 1;
            frame_send.ack = 0;
            printf("Enter data : ");
            scanf("%s",buffer);
            strcpy(frame_send.packet.data,buffer);

            //send just like udp program
            sendto(clientsocket,&frame_send,sizeof(frame_send),0,(struct
sockaddr*)&serveraddr,sizeof(serveraddr));
            printf("Frame send\n");
        }

        //recv just like udp program
        int frecv_size = recvfrom(clientsocket,&frame_recv,sizeof(frame_recv),0,(struct
sockaddr*)&serveraddr,&len);

        if(frecv_size > 0 && frame_recv.sq_no == 0 && frame_recv.ack == frameid+1){
            printf("Ack Received\n");
            ack_recv = 1;
        }
        else{
            printf("Ack not received\n");
            ack_recv = 0;
        }
        frameid++;
    }
    //close
    close(clientsocket);
}
```