

```
// server
```

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<arpa/inet.h>
#include<time.h>
#include<string.h>

void main()
{
    int sock=socket(AF_INET,SOCK_DGRAM,0);
    if(sock<0)
    {
        printf("Socket Creation Failed!!!\n");
        return;
    }

    struct sockaddr_in server_addr;
    server_addr.sin_family=AF_INET;
    server_addr.sin_port=htons(8000);
    server_addr.sin_addr.s_addr=inet_addr("127.0.0.1");
    if(bind(sock,(struct sockaddr*)&server_addr,sizeof(server_addr))<0)
    {
        printf("Binding Failed!!!\n");
        close(sock);
        return;
    }
    printf("Listening. .. \n");
    while(1)
    {
        char buffer[50];
        struct sockaddr_in client_addr;
        int addr_len=sizeof(client_addr);
        recvfrom(sock,buffer,sizeof(buffer),0,(struct sockaddr*)&client_addr,&addr_len);

        time_t cur_time=time(NULL);
        snprintf(buffer,sizeof(buffer),"%s",ctime(&cur_time));

        sendto(sock,buffer,strlen(buffer),0,(struct sockaddr*)&client_addr,addr_len);
        printf("Current Time: %sSent Succesfully.\n",buffer);
    }
    close(sock);
    return;
}
```

Client.c

```
#include<stdio.h>
#include<stdlib.h>
```

```

#include<unistd.h>
#include<arpa/inet.h>
#include<string.h>

void main()
{
    int sock = socket(AF_INET, SOCK_DGRAM, 0);
    if (sock<0)
    {
        printf("Socket Creation Failed!!!\n");
        return;
    }
    struct sockaddr_in server_addr;
    server_addr.sin_family=AF_INET;
    server_addr.sin_port=htons(8000);
    server_addr.sin_addr.s_addr=inet_addr("127.0.0.1");

    sendto(sock,"Time?",strlen("Time?"),0,(struct
sockaddr*)&server_addr,sizeof(server_addr));

    char buffer[50];
    int addr_len=sizeof(server_addr);
    memset(buffer,0,sizeof(buffer));
    recvfrom(sock,buffer,sizeof(buffer),0,(struct sockaddr*)&server_addr,&addr_len);

    printf("Current Time: %sReceived from server\n",buffer);
    close(sock);
    return;
}

```

OUTPUT

```

(Arun@kali)-[~/Desktop/network/cycle1]
$ ./server
Listening....
Current Time: Sat May 17 10:05:32 2025
Sent Succesfully.

```

```

(Arun@kali)-[~/Desktop/network/cycle1]
$ ./Client
Current Time: Sat May 17 10:05:32 2025
Received from server

```

```
// server
```

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

void handle_client(int client_socket)
{
    char buffer[50];
    int word_count=0;
    memset(buffer, 0, sizeof(buffer));
    if((recv(client_socket,&buffer,sizeof(buffer),0))<0)
    {
        printf("Receive Failed!!!\n");
        close(client_socket);
        return;
    }
    printf("%s\n",buffer);
    if(strcmp(buffer,"start")==0)
    {
        printf("Server communication Started!!\n");
        send(client_socket,"Server communication Started!!\n",strlen("Server communication
Started!!\n"),0);
    }
    else
    {
        printf("Invalid Start command!!\n");
        send(client_socket,"Invalid Start command!!\n",strlen("Invalid Start
command!!\n"),0);
        close(client_socket);
        exit(0);
    }
    while(1)
    {
        memset(buffer,0,sizeof(buffer));
        if((recv(client_socket,&buffer,sizeof(buffer),0))<0)
        {
            printf("Receive Failed!!!\n");
            close(client_socket);
            return;
        }
        buffer[strcspn(buffer,"\n")]=0;
        if(strcmp(buffer,"stop")==0)
        {
            send(client_socket,"Connection Terminated\n",strlen("Connection Terminated\n"),0);
            close(client_socket);
        }
    }
}
```

```

return;
}
int words=0;
char *token=strtok(buffer, " ");
while(token!=NULL)
{
words++;
token=strtok(NULL, " ");
}
if((word_count+words)>25)
{
send(client_socket,"Max limit Exceeded!!!\n",strlen("Max limit Exceeded!!!\n"),0);
close(client_socket);
return;
}
else{
word_count+=words;
send(client_socket,"Message Received\n",strlen("Message Received\n"),0);
}
}
close(client_socket);
return;
}

```

```

void main()
{
int sock=socket(AF_INET,SOCK_STREAM,0);
if(sock<0)
{
printf("Socket Creation Failed!!!\n");
return;
}
struct sockaddr_in server_addr,client_addr;
server_addr.sin_family=AF_INET;
server_addr.sin_port=htons(8000);
server_addr.sin_addr.s_addr=inet_addr("127.0.0.1");

if(bind(sock,(struct sockaddr*)&server_addr,sizeof(server_addr))<0)
{
printf("Binding Failed!!\n");
close(sock);
return;
}
printf("Binding Succesful!!\n");

if((listen(sock,5))<0)
{
printf("Listening Failed!!\n");
close(sock);
return;
}
}

```

```

    }
    printf("Listening...\n");

    while(1)
    {
        int addr_size=sizeof(client_addr);
        int client_socket=accept(sock,(struct sockaddr*)&client_addr,&addr_size);
        if(client_socket<0)
        {
            printf("Waiting for new connections. .. \n");
            sleep(5);
            continue;
        }
        printf("New Client Accepted!!\n");

        if(fork()==0)
        {
            close(sock);
            handle_client(client_socket);
        }
        close(client_socket);
    }
    close(sock);
    return;
}

```

Client.c

```

#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<arpa/inet.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<string.h>

void main()
{
    int sock = socket(AF_INET, SOCK_STREAM, 0);
    char buffer[50];
    if(sock<0)
    {
        printf("Socket Creation Failed!!!\n");
        return;
    }
    struct sockaddr_in server_addr;
    server_addr.sin_family=AF_INET;
    server_addr.sin_port=htons(8000);
    server_addr.sin_addr.s_addr=inet_addr("127.0.0.1");
}

```

```

if(connect(sock,(struct sockaddr*)&server_addr,sizeof(server_addr))<0)
{
printf("Connection Failed!!!\n");
close(sock);
return;
}

printf("Connected to Server\n");

send(sock,"start",strlen("start"),0);
recv(sock,buffer,sizeof(buffer),0);
printf("%s",buffer);

while(1)
{
if(strcmp(buffer,"Max limit Exceeded!!!\n")==0)
{
break;
}
printf("Enter your message(stop to exit) :");
fgets(buffer,sizeof(buffer),stdin);

send(sock,buffer,sizeof(buffer),0);

if(strcmp(buffer,"stop\n")==0)
{
break;
}
memset(buffer,0,sizeof(buffer));
recv(sock,buffer,sizeof(buffer),0);
printf("%s",buffer);
}
close(sock);
return;
}

```

OUTPUT

```

└─(Arun@kali)-[~/Desktop/network/cycle1]
└─$ ./server
Binding Succesful!!
Listening...
New Client Accepted!!
start
Server communication Started!!
Waiting for new connections....
Waiting for new connections....

```

```
(Arun@kali)-[~/Desktop/network/cycle1]
└─$ ./Client
Connected to Server
Server communication Started!!
Enter your message(stop to exit) :hi
Message Received
Enter your message(stop to exit) :helllo
Message Received
Enter your message(stop to exit) :what
Message Received
Enter your message(stop to exit) :2
Message Received
Enter your message(stop to exit) :3
Message Received
Enter your message(stop to exit) :Arun
Message Received
Enter your message(stop to exit) :hello worlds
Message Received
Enter your message(stop to exit) :stop
```