

# UCS1511 - COMPUTER NETWORKS

## SINGLE SERVER MULTI CLIENT PROGRAM

**REG NO : 205001085**

**EX.NO : 5**

**NAME : SABARIVASAN V**

**DATE: 14.09.22**

---

### **OBJECTIVE :**

To create a single server multi client chat application.

### **CODE :**

#### **CLIENT :**

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

#define PORT 9876

int main(int argc, char* argv[]){

    int clientSocket, ret;
    struct sockaddr_in serverAddr;

    clientSocket = socket(AF_INET, SOCK_STREAM, 0);
    if(clientSocket < 0){
        printf("[-]Error in connection.\n");
        exit(1);
    }
}
```

```

printf("[+]Client Socket is created.\n");

memset(&serverAddr, '\0', sizeof(serverAddr));
serverAddr.sin_family = AF_INET;
serverAddr.sin_port = htons(PORT);
serverAddr.sin_addr.s_addr = inet_addr("127.0.0.1");

ret = connect(clientSocket, (struct sockaddr*)&serverAddr, sizeof(serverAddr));
if(ret < 0){
    printf("[-]Error in connection.\n");
    exit(1);
}
printf("[+]Connected to Server.\n");

while(1){char buffer[1024];
    printf("%s: \t",argv[1]);
    scanf("%s", &buffer[0]);
    send(clientSocket, buffer, strlen(buffer), 0);
    bzero(buffer, sizeof(buffer));

    if(strcmp(buffer, ":exit") == 0){
        close(clientSocket);
        printf("[-]Disconnected from server.\n");
        exit(1);
    }

    if(recv(clientSocket, buffer, 1024, 0) < 0){
        printf("[-]Error in receiving data.\n");
    }else{
        printf("Server: \t%s\n", buffer);
    }
}

return 0;
}

```

## **SERVER:**

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

#define PORT 9856

int main(){

    int sockfd, ret;
    struct sockaddr_in serverAddr;

    int newSocket;
    struct sockaddr_in newAddr;

    socklen_t addr_size;

    pid_t childpid;

    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if(sockfd < 0){
        printf("[-]Error in connection.\n");
        exit(1);
    }
    printf("[+]Server Socket is created.\n");

    memset(&serverAddr, '\0', sizeof(serverAddr));
    serverAddr.sin_family = AF_INET;
    serverAddr.sin_port = htons(PORT);
    serverAddr.sin_addr.s_addr = inet_addr("127.0.0.1");
```

```

ret = bind(sockfd, (struct sockaddr*)&serverAddr, sizeof(serverAddr));
if(ret < 0){
    printf("[-]Error in binding.\n");
    exit(1);
}
printf("[+]Bind to port %d\n", PORT);

if(listen(sockfd, 10) == 0){
    printf("[+]Listening....\n");
}else{
    printf("[-]Error in binding.\n");
}

while(1){
    newSocket = accept(sockfd, (struct sockaddr*)&newAddr, &addr_size);
    if(newSocket < 0){
        exit(1);
    }
    printf("Connection accepted from %s:%d\n", inet_ntoa(newAddr.sin_addr),
    ntohs(newAddr.sin_port));

    if((childpid = fork()) == 0){
        close(sockfd);

        while(1){char buffer[1024];
            recv(newSocket, buffer, 1024, 0);
            if(strcmp(buffer, ":exit") == 0){
                printf("Disconnected from %s:%d\n",
inet_ntoa(newAddr.sin_addr), ntohs(newAddr.sin_port));
                break;
            }else{
                printf("Client: %s\n", buffer);
                bzero(buffer, sizeof(buffer));
                printf("Server: \t");
                scanf("%s", &buffer[0]);
                send(newSocket, buffer, strlen(buffer), 0);
                bzero(buffer, sizeof(buffer));
            }
        }
    }
}

```

```

    }

}

close(newSocket);

return 0;
}

```

## OUTPUT :

The image displays four terminal windows showing the execution of a multi-user chat application. The top-left window shows the server compilation and execution, which listens on port 9876 and handles multiple clients. The top-right window shows the compilation and execution of a client named 'sabari'. The bottom-left window shows the execution of a client named 'krishna'. The bottom-right window shows the execution of a client named 'selvi'. Each client window shows a successful connection to the server and a chat session where the user's input is echoed back.

```

root@spl16: ~/Desktop/sabari/multi user chat actual
root@spl16:~/Desktop/sabari/multi user chat actual# gcc server.c -o s
root@spl16:~/Desktop/sabari/multi user chat actual# ./s
[+]Server Socket is created.
[+]Bind to port 9876
[+]Listening....
Connection accepted from 127.0.0.1:42174
Connection accepted from 127.0.0.1:42178
Connection accepted from 127.0.0.1:42180
Client: sabarivasan
Server:      hi
Client: srikrishnan
Server:      vanakkam
Client: selviparasakthi
Server:      namaste
Disconnected from 127.0.0.1:42174
Disconnected from 127.0.0.1:42178
Disconnected from 127.0.0.1:42180
[]

Network
root@spl16: ~/Desktop/sabari/multi user chat actual
root@spl16:~/Desktop/sabari/multi user chat actual# ./c krishna
[+]Client Socket is created.
[+]Connected to Server.
krishna:      srikrishnan
Server:      vanakkam
krishna:      :exit
[]

root@spl16: ~/Desktop/sabari/multi user chat actual
root@spl16:~/Desktop/sabari/multi user chat actual# gcc client.c -o c
root@spl16:~/Desktop/sabari/multi user chat actual# ./c sabari
[+]Client Socket is created.
[+]Connected to Server.
sabari:      sabarivasan
Server:      hi
sabari:      :exit
[]

root@spl16: ~/Desktop/sabari/multi user chat actual
root@spl16:~/Desktop/sabari/multi user chat actual# ./c selvi
[+]Client Socket is created.
[+]Connected to Server.
selvi:      selviparasakthi
Server:      namaste
selvi:      :exit
[]

```