CN Lab- 2CS201CC24
SEM 5

INSTRUCTOR: DR. RAJESH GUPTA

Milap Patel, 22BCE186

13 August 2024

Lab 3: Develop Echo Client-server using socket programming

What is a socket?

- It is an abstraction through which an application may send and receive data. It provides generic access to interprocess communication services.
- Uniquely identified by an internet address, an end-to-end protocol (e.g. TCP or UDP), a port number
- There’re two types of TCP/IP sockets: stream socket (uses TCP) and datagram socket (uses UDP)

Client-Server communication

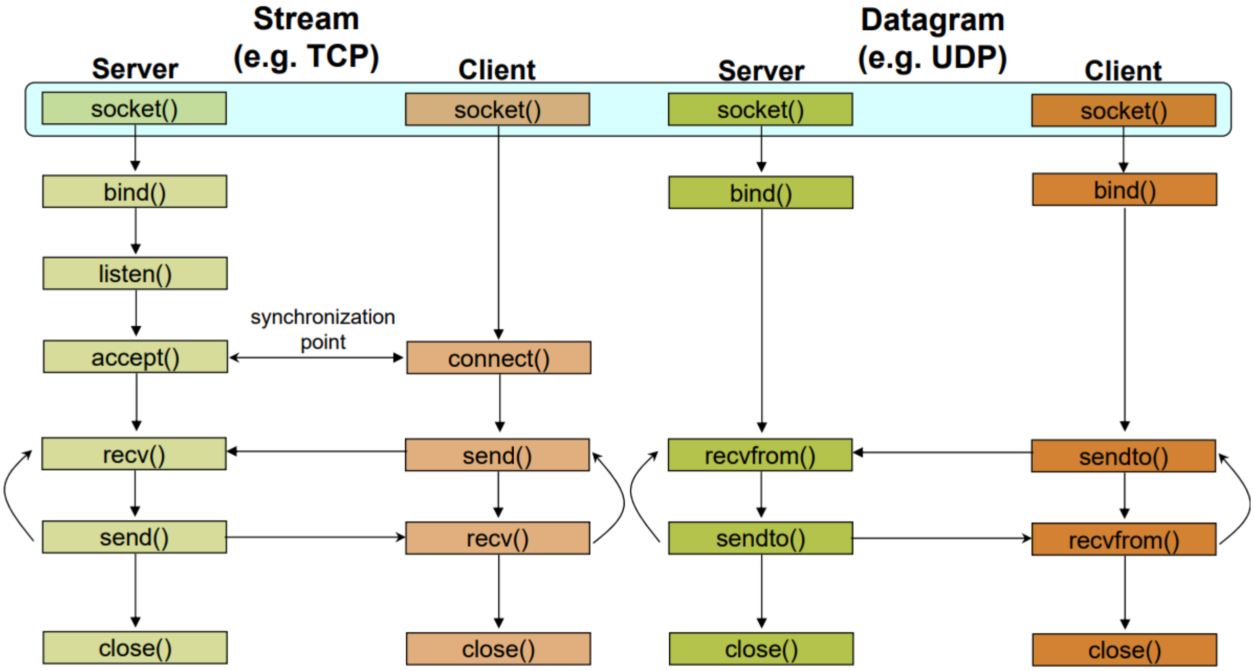
Server

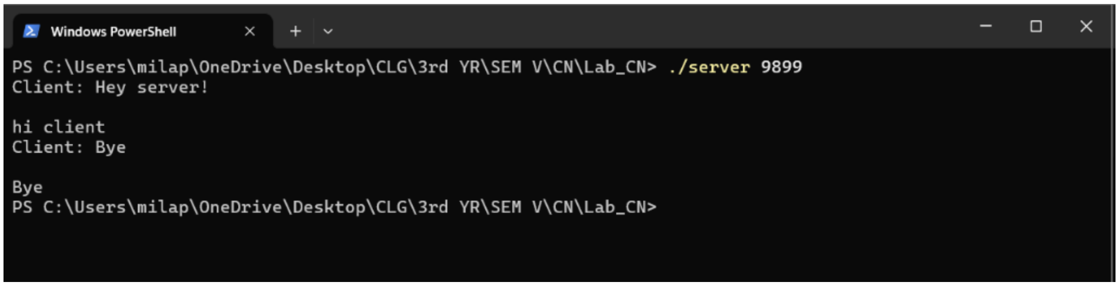
- ‰ passively waits for and responds to clients
- ‰ passive socket

Client

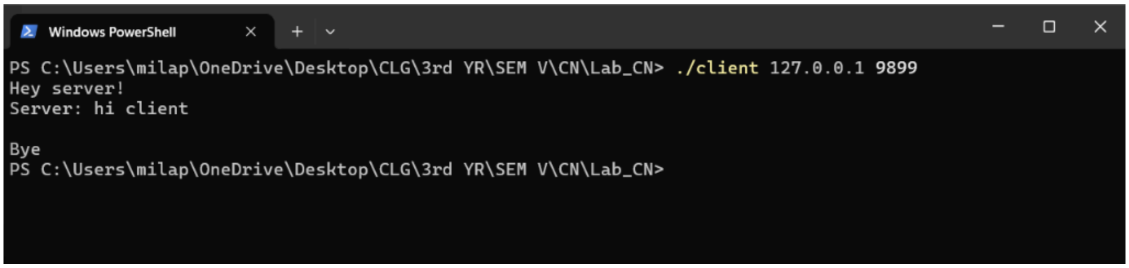
- ‰ initiates the communication
- ‰ must know the address and the port of the server active socket

Primitive	Meaning
Socket	Create a new communication endpoint
Bind	Attach a local address to a socket
Listen	Announce willingness to accept connections
Accept	Block caller until a connection request arrives
Connect	Actively attempt to establish a connection
Send	Send some data over the connection
Receive	Receive some data over the connection
Close	Release the connection

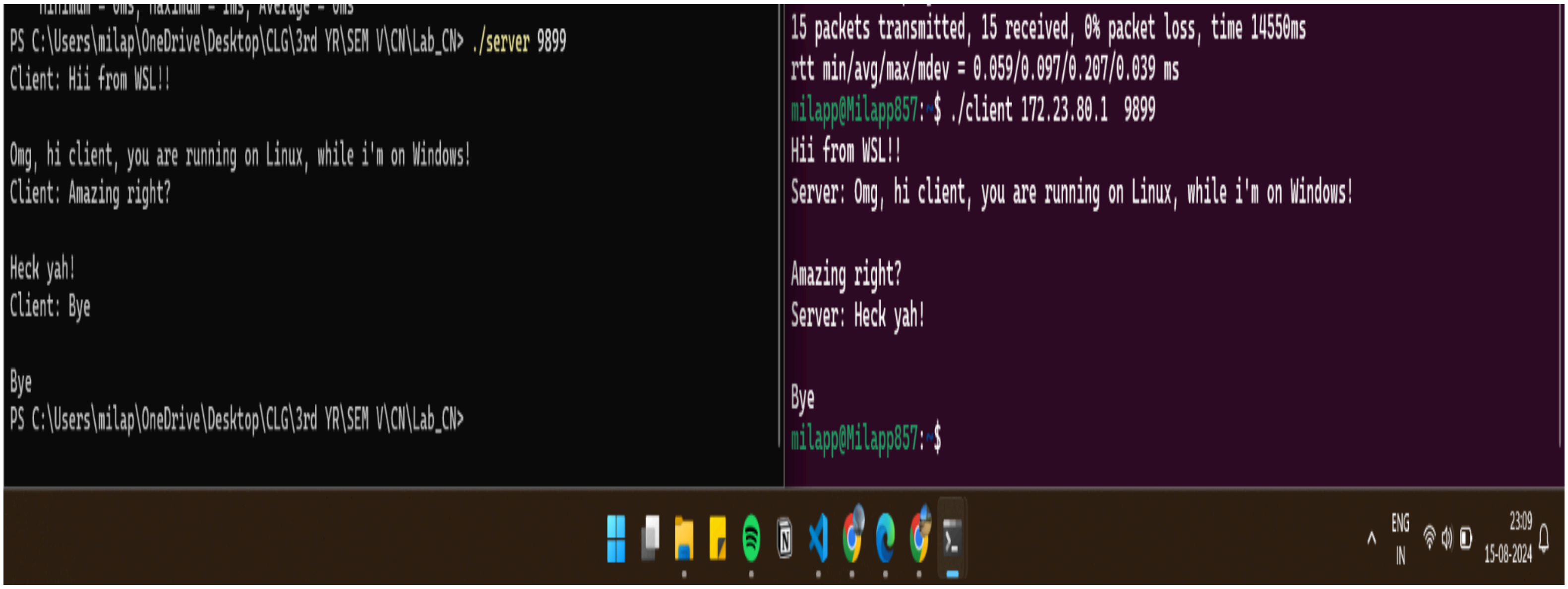




```
PS C:\Users\milap\OneDrive\Desktop\CLG\3rd YR\SEM V\CN\Lab_CN> ./server 9899
Client: Hey server!

hi client
Client: Bye

Bye
PS C:\Users\milap\OneDrive\Desktop\CLG\3rd YR\SEM V\CN\Lab_CN>
```



```
PS C:\Users\milap\OneDrive\Desktop\CLG\3rd YR\SEM V\CN\Lab_CN> ./client 127.0.0.1 9899
Hey server!
Server: hi client

Bye
PS C:\Users\milap\OneDrive\Desktop\CLG\3rd YR\SEM V\CN\Lab_CN>
```



```
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
PS C:\Users\milap\OneDrive\Desktop\CLG\3rd YR\SEM V\CN\Lab_CN> ./server 9899
Client: Hii from WSL!!

Omg, hi client, you are running on Linux, while i'm on Windows!
Client: Amazing right?

Heck yah!
Client: Bye

Bye
PS C:\Users\milap\OneDrive\Desktop\CLG\3rd YR\SEM V\CN\Lab_CN>
```

```
15 packets transmitted, 15 received, 0% packet loss, time 14550ms
rtt min/avg/max/mdev = 0.059/0.097/0.207/0.039 ms
milapp@Milapp857:~$ ./client 172.23.80.1  9899
Hii from WSL!!
Server: Omg, hi client, you are running on Linux, while i'm on Windows!

Amazing right?
Server: Heck yah!

Bye
milapp@Milapp857:~$
```

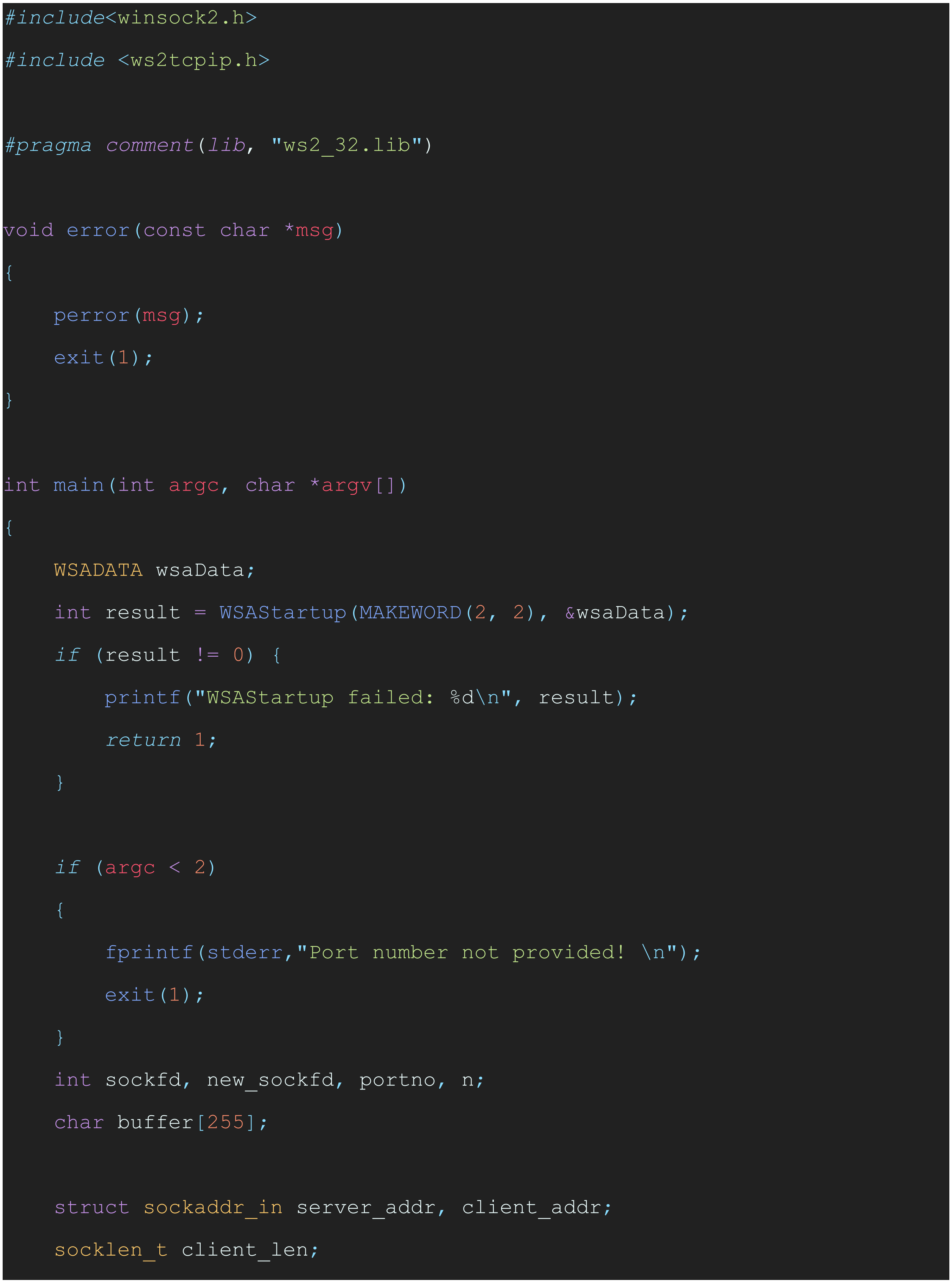
Code

Client (Linux)

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

void error(const char *msg)
{
    perror(msg);
    exit(1);
}

int main(int argc, char *argv[])
{
    if (argc < 3)
    {
        fprintf(stderr, "usage %s hostname port\n", argv[0]);
        exit(1);
    }

    int sockfd, portno, n;
    char buffer[255];
    portno = atoi(argv[2]);

    struct sockaddr_in server_addr;
    struct hostent *server;

    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0)
    {
        error("Error opening socket");
    }

    server = gethostbyname(argv[1]);
    if (server == NULL)
    {
        fprintf(stderr, "Error, no such host\n");
        exit(0);
    }

    memset(&server_addr, 0, sizeof(server_addr));
    server_addr.sin_family = AF_INET;
    memcpy((char *)&server_addr.sin_addr.s_addr, (char *)server->h_addr,
server->h_length);
    server_addr.sin_port = htons(portno);

    int connect_status = connect(sockfd, (struct sockaddr *)&server_addr,
sizeof(server_addr));
    if (connect_status < 0)
    {
        error("Connect failed");
    }

    while (1)
    {
        memset(&buffer, 0, sizeof(buffer));
        fgets(buffer, 255, stdin);
        n = write(sockfd, buffer, strlen(buffer));
        if (n < 0)
            error("Error on writing");

        int i = strncmp("Bye", buffer, 3);
        if (i == 0)
        {
            break;
        }

        memset(&buffer, 0, sizeof(buffer));
        n = read(sockfd, buffer, 255);
        if (n < 0)
            error("Error on reading");
        printf("Server: %s\n", buffer);
    }

    close(sockfd);
    return 0;
}
```

Server (Windows)

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



```c
#include<winsock2.h>

#include <ws2tcpip.h>


#pragma comment(lib, "ws2_32.lib")


void error(const char *msg)

{

    perror(msg);

    exit(1);

}


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

{

    WSADATA wsaData;

    int result = WSAStartup(MAKEWORD(2, 2), &wsaData);

    if (result != 0) {

        printf("WSAStartup failed: %d\n", result);

        return 1;

    }


    if (argc < 2)

    {

        fprintf(stderr,"Port number not provided! \n");

        exit(1);

    }

    int sockfd, new_sockfd, portno, n;

    char buffer[255];


    struct sockaddr_in server_addr, client_addr;

    socklen_t client_len;
    sockfd = socket(AF_INET, SOCK_STREAM,0);
    if (sockfd<0)
    {
        error("Error opening socket");
    }
    memset(&server_addr, 0, sizeof(server_addr));

    portno = atoi(argv[1]);
    //argv[0] -> filename and argv[1]->port name

    server_addr.sin_family = AF_INET;
    server_addr.sin_addr.s_addr = INADDR_ANY;
    server_addr.sin_port = htons(portno); //hton -> host to network short ?

    int binding_status = bind(sockfd, (struct sockaddr *)&server_addr, sizeof(server_addr));
    if(binding_status < 0){
        error("Binding failed.");
    }

    listen(sockfd, 5);
    client_len = sizeof(client_addr);

    new_sockfd = accept(sockfd, (struct sockaddr *)&client_addr, &client_len );
    if (new_sockfd<0)
    {
        error("Error on Accept");
    }

    while (1)
    {
        memset(&buffer, 0, sizeof(buffer));
        n = recv(new_sockfd, buffer, 255,0);
        if (n<0)
        {
            error("Error on reading!");
        }
        printf("Client: %s \n", buffer);

        memset(&buffer, 0, sizeof(buffer));
        fgets(buffer, 255, stdin);
        n = send(new_sockfd, buffer, strlen(buffer),0);
        if (n<0)
        {
            error("Error on writing!");
        }

        int i = strncmp("Bye",buffer,3);
        if (i==0)
        {
            break;
        }
    }
    closesocket(new_sockfd);
    closesocket(sockfd);
    WSACleanup();
    return 0;
}
```