 **<winsock2.h>**: Windows soket (ağ) programlaması için gerekli kütüphane. Ağ bağlantılarını kurmak için kullanılır.

 **<ws2tcpip.h>**: Modern Winsock fonksiyonlarını (örneğin IPv6 desteği) içerir.

 **<stdio.h>**: Ekrana yazı yazdırmak veya veri okumak için standart giriş-çıkış fonksiyonlarını içerir (ör. printf).

WSADATA wsaData;// Winsock Kütüphanesini Yükleme

if (WSAStartup(MAKEWORD(2, 2), &wsaData) != 0) {

printf("WSAStartup failed. Error: %d\n", WSAGetLastError());

return 1;

}

 **WSAStartup**: Winsock kütüphanesini başlatır.

 **MAKEWORD(2, 2)**: Winsock 2.2 sürümünü kullanmak istediğimizi belirtiyoruz.

 Eğer başlatma başarısız olursa, bir hata mesajı yazdırıp programı sonlandırıyoruz.

serverSocket = socket(AF\_INET, SOCK\_STREAM, 0);

if (serverSocket == INVALID\_SOCKET) {

printf("Socket creation failed. Error: %d\n", WSAGetLastError());

WSACleanup();

return 1;

}

 **socket**: Sunucu soketi (bağlantı noktası) oluşturur.

* **AF\_INET**: IPv4 protokolünü kullanır.
* **SOCK\_STREAM**: TCP bağlantısı kurmak için kullanılır (veri akışı).

 **INVALID\_SOCKET**: Eğer soket oluşturulamazsa bu hata döner.

Sunucunun Adresini Belirleme

serverAddr.sin\_family = AF\_INET;

serverAddr.sin\_addr.s\_addr = INADDR\_ANY;

serverAddr.sin\_port = htons(PORT);

Soketi Bağlama (Bind)

if (bind(serverSocket, (struct sockaddr\*)&serverAddr, sizeof(serverAddr)) == SOCKET\_ERROR) {

printf("Bind failed. Error: %d\n", WSAGetLastError());

closesocket(serverSocket);

WSACleanup();

return 1;

}

 **bind**: Sunucu soketini belirlediğimiz IP adresi ve porta bağlar.

 Eğer başarısız olursa, soketi kapatır ve Winsock'u temizleriz.

**Dinlemeye Başlama**

if (listen(serverSocket, 3) == SOCKET\_ERROR) {

printf("Listen failed. Error: %d\n", WSAGetLastError());

closesocket(serverSocket);

WSACleanup();

return 1;

}

printf("Server listening on port %d...\n", PORT);

**listen**: Sunucu, gelen bağlantıları dinlemeye başlar.

* **3**: Maksimum 3 istemciyi sırada bekletebilir.

Bağlantı Kabul Etme

clientSocket = accept(serverSocket, (struct sockaddr\*)&clientAddr, &addrLen);

if (clientSocket == INVALID\_SOCKET) {

printf("Accept failed. Error: %d\n", WSAGetLastError());

closesocket(serverSocket);

WSACleanup();

return 1;

}

printf("Connection established with client.\n");

 **accept**: İstemciden gelen bir bağlantıyı kabul eder.

* Bağlantı kurulduğunda, istemci ile iletişim kurmak için yeni bir soket döner.

 Veri Alma (Receive)

Eğer bağlantı kabul edilemezse hata verir.

int bytesReceived = recv(clientSocket, buffer, BUFFER\_SIZE, 0);

if (bytesReceived > 0) {

printf("Message from client: %s\n", buffer);

}

- recv: İstemciden veri alır ve bunu buffer içine yazar.

-BUFFER\_SIZE: Alınabilecek maksimum veri boyutu.

- Gelen mesaj ekrana yazdırılır

Bağlantıyı Kapatma ve Winsock'u Temizleme

closesocket(clientSocket);

closesocket(serverSocket);

WSACleanup();

**Programın Çalışma Mantığı**

1. Winsock başlatılır ve bir soket oluşturulur.
2. Soket bir IP adresine ve porta bağlanır.
3. Sunucu gelen bağlantıları dinlemeye başlar.
4. Bir istemci bağlandığında, bağlantı kabul edilir.
5. İstemciden mesaj alınır ve ekrana yazdırılır.
6. Bağlantılar kapatılır ve Winsock temizlenir.