**27. Implementation of a DNS server and client in java/C using UDP sockets.**

**Server:**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <string.h>**

**#include <winsock2.h>**

**#include <ws2tcpip.h>**

**#pragma comment(lib, "ws2\_32.lib") // Link Winsock library**

**#define PORT 8080**

**#define BUFFER\_SIZE 1024**

**int main() {**

**WSADATA wsa;**

**SOCKET sockfd;**

**struct sockaddr\_in server\_addr, client\_addr;**

**char buffer[BUFFER\_SIZE];**

**int addr\_len = sizeof(client\_addr);**

**// Initialize Winsock**

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

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

**return 1;**

**}**

**// Create UDP socket**

**if ((sockfd = socket(AF\_INET, SOCK\_DGRAM, 0)) == INVALID\_SOCKET) {**

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

**return 1;**

**}**

**// Configure server address**

**server\_addr.sin\_family = AF\_INET;**

**server\_addr.sin\_addr.s\_addr = INADDR\_ANY;**

**server\_addr.sin\_port = htons(PORT);**

**// Bind socket**

**if (bind(sockfd, (struct sockaddr\*)&server\_addr, sizeof(server\_addr)) == SOCKET\_ERROR) {**

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

**return 1;**

**}**

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

**while (1) {**

**// Receive domain name from client**

**recvfrom(sockfd, buffer, BUFFER\_SIZE, 0, (struct sockaddr\*)&client\_addr, &addr\_len);**

**printf("Received request for domain: %s\n", buffer);**

**struct hostent \*host;**

**struct in\_addr \*\*addr\_list;**

**char ip[INET\_ADDRSTRLEN];**

**// Get IP address from hostname**

**if ((host = gethostbyname(buffer)) == NULL) {**

**strcpy(ip, "Error: Unable to resolve domain");**

**} else {**

**addr\_list = (struct in\_addr\*\*)host->h\_addr\_list;**

**strcpy(ip, inet\_ntoa(\*addr\_list[0])); // Convert to string**

**}**

**// Send IP address back to client**

**sendto(sockfd, ip, strlen(ip), 0, (struct sockaddr\*)&client\_addr, addr\_len);**

**printf("Resolved IP: %s\n", ip);**

**}**

**closesocket(sockfd);**

**WSACleanup();**

**return 0;**

**}**

**Client:**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <string.h>**

**#include <winsock2.h>**

**#include <ws2tcpip.h>**

**#pragma comment(lib, "ws2\_32.lib") // Link Winsock library**

**#define SERVER\_IP "127.0.0.1"**

**#define PORT 8080**

**#define BUFFER\_SIZE 1024**

**int main() {**

**WSADATA wsa;**

**SOCKET sockfd;**

**struct sockaddr\_in server\_addr;**

**char domain[BUFFER\_SIZE], response[BUFFER\_SIZE];**

**int addr\_len = sizeof(server\_addr);**

**// Initialize Winsock**

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

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

**return 1;**

**}**

**// Create UDP socket**

**if ((sockfd = socket(AF\_INET, SOCK\_DGRAM, 0)) == INVALID\_SOCKET) {**

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

**return 1;**

**}**

**// Configure server address**

**server\_addr.sin\_family = AF\_INET;**

**server\_addr.sin\_port = htons(PORT);**

**server\_addr.sin\_addr.s\_addr = inet\_addr(SERVER\_IP);**

**// Get domain name from user**

**printf("Enter domain name: ");**

**scanf("%s", domain);**

**// Send domain name to server**

**sendto(sockfd, domain, strlen(domain), 0, (struct sockaddr\*)&server\_addr, addr\_len);**

**// Receive IP address from server**

**recvfrom(sockfd, response, BUFFER\_SIZE, 0, (struct sockaddr\*)&server\_addr, &addr\_len);**

**printf("Resolved IP: %s\n", response);**

**closesocket(sockfd);**

**WSACleanup();**

**return 0;**

**}**

**A screenshot of a computer

AI-generated content may be incorrect.**