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;
    }
```

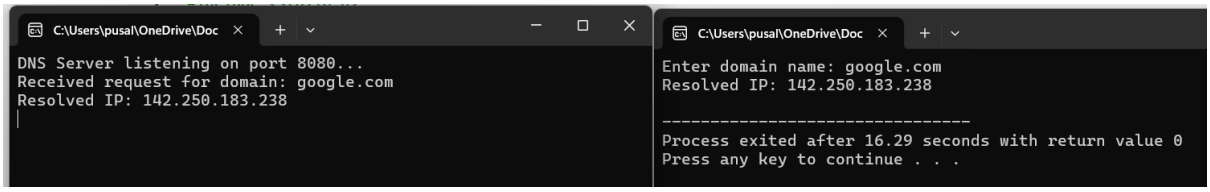
```
// Configure server address
server_addr.sin_family = AF_INET;
server_addr.sin_addr.s_addr = INADDR_ANY;
server_addr.sin_port = htons(PORT);

// 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;
}
```


C:\Users\pusal\OneDrive\Doc
DNS Server listening on port 8080...
Received request for domain: google.com
Resolved IP: 142.250.183.238
C:\Users\pusal\OneDrive\Doc
Enter domain name: google.com
Resolved IP: 142.250.183.238

Process exited after 16.29 seconds with return value 0
Press any key to continue . . .