SIT – 202 COMPUTER NETWORKS AND COMMUNICATION :

DNS SERVER SKETCH jasveena-224001588

PSEUDOCODE FOR DNS SERVER:

```
# Server Initialization
FUNCTION initialize_server():
    # Create a UDP socket for communication
    socket = create_udp_socket()

    # Bind the socket to the standard DNS port (93)
    bind_socket(socket, port=93)

    # Print confirmation message
    PRINT "DNS Server initialized and listening on port 93"

    # Load or initialize DNS records (A and CNAME records)
    load_dns_records()

    RETURN socket
END FUNCTION

# Function to create a UDP socket
FUNCTION create_udp_socket():
    RETURN CREATE_SOCKET(AF_INET, SOCK_DGRAM)
END FUNCTION

# Function to bind the socket to a specific port
FUNCTION bind_socket(socket, port):
    BIND(socket, '', port)
END FUNCTION
```

```
# Function to load DNS records
FUNCTION load_dns_records():
    # Initialize or load A and CNAME records from a file or database
    # For this example, records are initialized directly
    A_RECORDS = {
        "google.com": "93.184.216.34"
    }
    CNAME_RECORDS = {
        "www.google.com": "google.com"
    }
END FUNCTION
```

LISTENING AND PROCESSING DNS QUERIES :

```
# Listening and Processing DNS Queries
FUNCTION listen_for_queries(socket):
    WHILE TRUE:
        # Receive data from a client
        query, client_address = receive_query(socket)

        # Print the received query details
        PRINT "Received query from", client_address

        # Parse the DNS query to extract hostname and query type
        hostname, query_type = parse_query(query)

        # Print parsed query details
        PRINT "Parsed query: hostname =", hostname, ", query_type =", query_type
        # Handle the query based on its type
        IF query_type == "A":
            handle_a_record_query(hostname, client_address, socket)
        ELSE IF query_type == "CNAME":
            handle_cname_record_query(hostname, client_address, socket)
        ELSE:
            PRINT "Unsupported query type:", query_type
    END WHILE
END FUNCTION

# Function to receive data from the socket
FUNCTION receive_query(socket):
    RETURN RECEIVE_FROM(socket, BUFFER_SIZE)
END FUNCTION

# Function to parse the DNS query
FUNCTION parse_query(query):
    # Decode the query data to extract hostname and query type
    decoded_query = DECODE(query)
    hostname = EXTRACT_HOSTNAME(decoded_query)
    query_type = EXTRACT_QUERY_TYPE(decoded_query)

    RETURN hostname, query_type
END FUNCTION
```

HANDLING A and CNAME RECORDS:

```
# Handling A and CNAME Records
# Handle A record queries
FUNCTION handle_a_record_query(hostname, client_address, socket):
    # Look up the IP address for the given hostname
    ip_address = lookup_a_record(hostname)

    IF ip_address IS NOT NULL:
        # Generate a DNS response with the IP address
        response = generate_response(hostname, ip_address, query_type="A")
        send_response(response, client_address, socket)
    ELSE:
        PRINT "A record not found for hostname:", hostname
END FUNCTION

# Handle CNAME record queries
FUNCTION handle_cname_record_query(hostname, client_address, socket):
    # Look up the canonical name for the given hostname
    canonical_name = lookup_cname_record(hostname)

    IF canonical_name IS NOT NULL:
        # Generate a DNS response with the canonical name
        response = generate_response(hostname, canonical_name, query_type="CNAME")
        send_response(response, client_address, socket)
    ELSE:
        PRINT "CNAME record not found for hostname:", hostname
END FUNCTION

# Function to look up A record
FUNCTION lookup_a_record(hostname):
    RETURN A_RECORDS[hostname]
END FUNCTION
```

```
# Function to look up CNAME record
FUNCTION lookup_cname_record(hostname):
    RETURN CNAME_RECORDS[hostname]
END FUNCTION
```

GENERATING DNS RESPONSES :

```
# Generating DNS Responses
# Generate a DNS response message
FUNCTION generate_response(hostname, record_data, query_type):
    # Create a DNS response object
    response = create_dns_response()

    # Set hostname, record data, and query type in the response
    response.set_hostname(hostname)
    response.set_record_data(record_data)
    response.set_query_type(query_type)

    RETURN response
END FUNCTION

# Function to create a DNS response object
FUNCTION create_dns_response():
    RETURN NEW DNSResponse()
END FUNCTION

# Send the DNS response to the client
FUNCTION send_response(response, client_address, socket):
    # Send the response data to the client
    send_data(socket, response, client_address)
    PRINT "Sent response to", client_address
END FUNCTION


# Function to send data to the client
FUNCTION send_data(socket, response, client_address):
    SEND_TO(socket, response, client_address)
END FUNCTION
```

MAIN FUNCTION:

```
# Main function to start the DNS server
FUNCTION main():
    # Initialize the server
    socket = initialize_server()

    # Start listening for and processing queries
    listen_for_queries(socket)
END FUNCTION


# Start the DNS Server
main()
```

A LITTLE EXPLAINATION OF WHAT I DID:

Explanation

1. ***Server Initialization:***
    - ***initialize_server(): Sets up the UDP socket, binds it to port 93, and initializes DNS records.***
2. ***Listening and Processing DNS Queries:***

- ***listen_for_queries(socket): Continuously listens for incoming queries, processes them, and routes them based on the query type.***
- ***receive_query(socket): Receives a query from the client.***
- ***parse_query(query): Extracts the hostname and query type from the received query.***

3. ***Handling A and CNAME Records:***
    - ***handle_a_record_query(hostname, client_address, socket): Processes A record queries, looks up the IP address, and sends a response.***
    - ***handle_cname_record_query(hostname, client_address, socket): Processes CNAME record queries, looks up the canonical name, and sends a response.***
    - ***lookup_a_record(hostname): Looks up an A record.***
    - ***lookup_cname_record(hostname): Looks up a CNAME record.***
4. ***Generating DNS Responses:***
    - ***generate_response(hostname, record_data, query_type): Creates a DNS response based on the hostname, record data, and query type.***
    - ***send_response(response, client_address, socket): Sends the DNS response to the client.***
    - ***send_data(socket, response, client_address): Handles sending data to the client.***
5. ***Main Function:***
    - ***main(): Initializes the server and starts the process of listening and handling DNS queries.***