

c dns_server.c ●

Users > moshika > Desktop > c dns_server.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <unistd.h>
5 #include <arpa/inet.h>
6 #define PORT 8080
7 struct Record {
8     char domain[50];
9     char ip[20];};
10 struct Record dnsTable[] = {
11     {"google.com", "142.250.182.206"}, 
12     {"youtube.com", "142.250.181.110"}, 
13     {"microsoft.com", "20.81.111.85"}, 
14     {"example.com", "93.184.216.34"}};
15 int main() {
16     int sockfd;
17     char buffer[100];
18     struct sockaddr_in serverAddr, clientAddr;
19     socklen_t addr_size;
20     sockfd = socket(AF_INET, SOCK_DGRAM, 0);
21     serverAddr.sin_family = AF_INET;
22     serverAddr.sin_port = htons(PORT);
23     serverAddr.sin_addr.s_addr = INADDR_ANY;
24     bind(sockfd, (struct sockaddr *)&serverAddr, sizeof(serverAddr));
25     printf("DNS Server running...\n");
26     while (1) {
27         addr_size = sizeof(clientAddr);
28         recvfrom(sockfd, buffer, sizeof(buffer), 0,
29                   (struct sockaddr *)&clientAddr, &addr_size);
30         buffer[strcspn(buffer, "\n")] = 0;
31         printf("Query received: %s\n", buffer);
32         char result[50] = "NOT FOUND";
33         for (int i = 0; i < 4; i++) {
34             if (strcmp(buffer, dnsTable[i].domain) == 0) {
35                 strcpy(result, dnsTable[i].ip);
36                 break;}}
37         sendto(sockfd, result, strlen(result) + 1, 0,
38                (struct sockaddr *)&clientAddr, addr_size);
39     }
40     close(sockfd);
41     return 0;
42 }
```

c dns_server.c ● c dns_client.c ×

Users > moshika > Desktop > c dns_client.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <arpa/inet.h>
5 #include <unistd.h>
6
7 #define PORT 8080
8
9 int main() {
10     int sockfd;
11     char buffer[100], domain[50];
12     struct sockaddr_in serverAddr;
13     socklen_t addr_size;
14
15     sockfd = socket(AF_INET, SOCK_DGRAM, 0);
16
17     serverAddr.sin_family = AF_INET;
18     serverAddr.sin_port = htons(PORT);
19     serverAddr.sin_addr.s_addr = inet_addr("127.0.0.1");
20
21     printf("Enter domain name: ");
22     scanf("%s", domain);
23
24     sendto(sockfd, domain, strlen(domain) + 1, 0,
25             (struct sockaddr *)&serverAddr, sizeof(serverAddr));
26
27     addr_size = sizeof(serverAddr);
28     recvfrom(sockfd, buffer, sizeof(buffer), 0,
29               (struct sockaddr *)&serverAddr, &addr_size);
30
31     printf("IP Address: %s\n", buffer);
32
33     close(sockfd);
34     return 0;
35 }
```

Desktop — dns_server — 80x24

```
Last login: Sat Dec  6 23:41:34 on ttys000
moshika@Moshikas-MacBook-Air ~ % cd desktop
moshika@Moshikas-MacBook-Air desktop % gcc dns_server.c -o dns_server

moshika@Moshikas-MacBook-Air desktop % gcc dns_client.c -o dns_client
moshika@Moshikas-MacBook-Air desktop % ./dns_server
```

DNS Server running...

Query received: google.com

 Desktop --zsh-- 80x24

```
Last login: Sat Dec  6 23:42:15 on ttys000
moshika@Moshikas-MacBook-Air ~ % cd desktop
moshika@Moshikas-MacBook-Air desktop % ./dns_client
Enter domain name: google.com
IP Address: 142.250.182.206
moshika@Moshikas-MacBook-Air desktop % █
```