Department of Information and Communication Technology
Network Programming

C Program to resolve domain names

Prepared by: NGUYEN Hoang Minh - BI11-181
Instructor: TRAN Giang Son
Date: May 26, 2022

Contents

1 Problem Analysis

1.1 Get domain name from user

- Input domain name from CLI arguments
- Input domain name from user

1.2 Resolve domain name

- Hostname is not found
- Hostname is resolved successfully with multiple IPs (if possible)

2 Get domain name from user

Initially, a char array of length 255 is declared to store the domain name

```
char input[255];
```

2.1 Domain name comes from program's arguments

`argc` is the argument count. If `argc` is greater than one, there is at least one argument. We copy the second argument of `argv` (the first argument is the program name) into the `domain` char array

```
if (argc > 1)
{
    strcpy(input, argv[1]);
}
```

2.2 Domain name comes from user input

If no argument is provided, we prompt the user to enter a domain name and store the input into the `domain` char array

```
else
{
    printf("Enter a domain name: ");
    fgets(input, sizeof(input), stdin);
    input[strlen(input) - 1] = '\0';
}
```

3 Resolve domain name

Utilizing the function `gethostbyname()`, we can resolve the `domain` and obtain either a pointer to a `struct hostent` or a null pointer.

3.1 Hostname is not found

If we obtain a null pointer, the domain name cannot be resolved. Then we inform the user and exit the program

```
if (host_info == NULL)
{
    printf("No such host\n");
    exit(1);
}
```

3.2 Hostname is resolved successfully

If we obtain a pointer to a `struct hostent`, we loop through its address list, convert each address to a char array using the function `inet_ntop()` and print the result

```
for (int i = 0; host_info->h_addr_list[i] != NULL; i++)
{
    struct in_addr *address;
    address = (struct in_addr *)(host_info->h_addr_list[i]);

    printf("%s has address %s\n", input, inet_ntoa(*address));
}
```

4 Demonstration

Compile the program to an output file name `mynslookup`

4.1 Run the program locally

```
> ./mynslookup stackoverflow1.com
No such host

> ./mynslookup
Enter a domain name: facebook.com
facebook.com has address 31.13.75.35


> ./mynslookup stackoverflow.com
stackoverflow.com has address 151.101.1.69
stackoverflow.com has address 151.101.193.69
stackoverflow.com has address 151.101.129.69
stackoverflow.com has address 151.101.65.69
```

4.2 Run the program on a VPS in Singapore

```
> ./mynslookup stackoverflow1.com
No such host

> ./mynslookup
Enter a domain name: facebook.com
facebook.com has address 157.240.235.35


> ./mynslookup stackoverflow.com
stackoverflow.com has address 151.101.1.69
stackoverflow.com has address 151.101.193.69
stackoverflow.com has address 151.101.129.69
stackoverflow.com has address 151.101.65.69
```

The resolved IP address of facebook.com on my local machine `31.13.75.35` is different from the one from my VPS `157.240.235.35` because one domain name can map to multiple IP addresses and the geological distance can cause the mapping to behave differently in diffent region.