

NP-Lab DNS Server

Assignment

必须实现域名系统客户端 DNS client 和 DNS Server 服务器。这意味着该协议将在最低工作水平上实施。允许以下限制：

- 仅支持 A 类记录。
- 服务器地址必须等于请求中域名的长度。例如 `ya.ru. = 0.0.0.6` 和 `test.ru. = 0.0.0.8`
- 不需要在答案中指定权威服务器或附加记录。只（需要）直接回答。

Description server options:

```
_____ server cli _____  
1 ./server [port]  
2 port by default 53  
_____
```

Description client options:

```
_____ client cli _____  
1 ./client dns_name [server] [port]  
2 dns_name mandatory option  
3 server by default localhost  
4 port by default 53  
_____
```

默认情况下，实用程序必须使用 53 个端口。为了节省时间，请使用嗅探器捕获的包作为查询/响应模板。Wireshark 为此提供了导出为 C 数组。要测试 DNS 服务器，请使用 `host`、`dig` 或 `nslookup` 控制台，明确告诉它们在同一台计算机上访问服务器。使用任何现有的域名系统服务器，如 Yandex 域名系统(77.88.8.8)来验证您的域名系统客户端。此外，还有许多关于域名系统协议的简化描述。

默认情况下，实用程序必须使用端口 53。为了节省时间，请使用 `sniffer` 捕获的包作为查询/响应模板。Wireshark 为此提供了为 C 数组的导出。

要测试 DNS 服务器，请使用 `host`、`dig` 或 `nslookup` 控制台，明确告诉它们在同一台计算机上访问服务器。使用任何现有的域名系统服务器，如 Yandex 域名系统(77.88.8.8)来验证您的域名系统客户端。

[RFC DNS](#) 此外，还有许多对 DNS 协议的简化描述。

Guidance

DNS 任务必须实现 DNS 客户端和服务端。这意味着该协议将在最低工作水平上实施。允许以下限制：

- 仅支持 A 类型记录
- 服务器可以提供任何地址作为响应。它可以是域名散列、来自文件(例如/etc/hosts)的地址、其长度等于请求中域名长度的带有种子的随机值或您选择的其他内容。唯一的希望是返回的地址是请求中的域名函数和本地计算机上的指定文件，并且不依赖于其他值(以确保请求的幂等性)。还可以实现一个使用 `getaddrinfo / gethostbyname` 调用以从真实的 DNS 网络请求地址的服务器。
- 没有必要在答案中指定权威服务器或附加记录。只（需要）直接回答。

To save time, use a sniffer-captured package as a query/response template.

Wireshark has Export as C Arrays for this purpose.

To test a DNS server, use `host`, `dig`, or `nslookup` consoles, explicitly telling them to access the server on the same computer. Use any existing DNS server, such as Yandex DNS (77.88.8.8), to verify your DNS client.

RFC DNS In addition, there are many simplified descriptions of the DNS protocol.

一些有用的评论

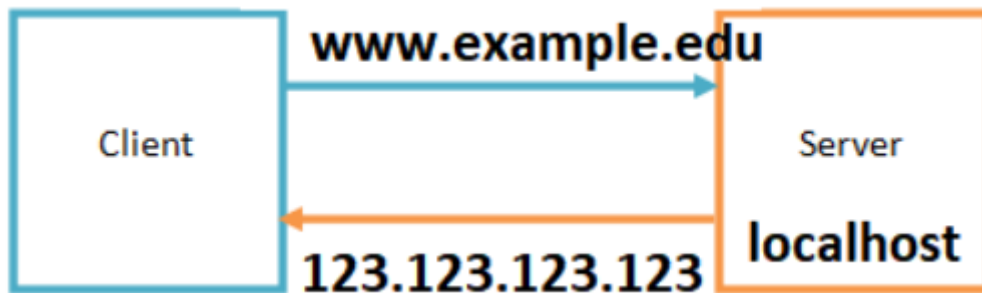
DNS(域名系统)- 将有意义的名称 (称为主机名) 如 `http://www.microsoft.com` 链接到特定 IP 地址(如 192.168.124.1)的数据库。

任务是创建一个 `server.c` 文件和一个 `client.c` 文件。`client.c` 文件应该从命令行接受参数: dns 服务器和解析地址(如 `www.example.com`)。因此，客户端应该向服务器发送一条消息，以便获取解析地址的 IP 地址

在此分配中，您不需要创建用于存储主机名和 IP 地址对的数据库，相反，您可以实施以下策略之一来获取地址(地址不一定是真实的，它可以是随机值)：

- a domain name hash,
- an address from a file (e.g./etc/hosts) on your computer,
- a random value with a seed equal to the length of a domain name in a request
- use getaddrinfo / gethostbyname calls to request addresses from a real DNS network.

您的目标是创建最简单的 DNS 服务器来处理一个客户端，因此不要使用线程。交流方案如下图所示：



为了您可以方便地与服务器通信，您将获得一个 `dns_protocol` 头文件。该协议的思想是，您应该填充 `dns_protocol.h` 中描述的给定结构(只有必要的字段)，然后将它们放入缓冲区，并通过 UDP 将该缓冲区发送到服务器。服务器应该解析消息并获得关于 `question()` 的适当信息，因为它知道 `dns_protocol.h` 中的结构。然后它应该以某种方式解析地址并将其发送给客户端。仅此而已。

下面是如何在缓冲区中放置结构和其他有用信息的示例：

```

struct DNS_HEADER *dns = (struct DNS_HEADER*)buffer; //
    unsigned char *qname = (unsigned char*)(buffer +
sizeof(struct DNS_HEADER));
    dns->qr = 1;
    dns->ans_count = htons(1);
    unsigned char* reader = buffer + sizeof(struct DNS_HEADER) +
strlen((const char*)qname) + 1 + sizeof(struct QUESTION);
  
```