

北京邮电大学课程设计报告

课程设计 名称	计算机网络课程 设计	学 院	计算机	指导教师	程莉
班 级	班内序号	学 号	学生姓名	成绩	
2020211323	11	2020211429	陈宇		
课 程 设 计 内 容	<p>课程设计要求设计一个 DNS 服务器程序，读入“IP 地址-域名”对照表，当客户端查询域名对应的 IP 地址时，用域名检索该对照表，有三种可能检索结果：</p> <p>1) 检索结果：ip 地址 0.0.0.0，则向客户端返回“域名不存在”的报错消息（不良网站拦截功能）</p> <p>2) 检索结果：普通 IP 地址，则向客户端返回该地址（服务器功能）</p> <p>3) 表中未检测到该域名，则向因特网 DNS 服务器发出查询，并将结果返给客户端（中继功能）</p> <p>课程设计一个人独立完成。</p>				
学生 课程设计 报告 (附页)					
课 程 设 计 成 绩 评 定	<p>评语：</p> <p>成绩：</p> <p>指导教师签名：</p> <p>年 月 日</p>				

注：评语要体现每个学生的工作情况，可以加页。

一、系统的功能设计

1、基本功能：开启 DNS 服务器时，读入“域名-IP 地址”对照表，当客户端查询域名对应的 IP 地址时，用域名检索该对照表。若结果为 IP 地址 0.0.0.0，则向客户端返回 NAME ERROR 报错信息；若结果为普通 IP 地址，则向客户返回这个地址；若表中未找到该域名，则向主机 DNS 服务器发出查询，并将结果返回给客户端。

2、多客户端并发：采用多线程结构，DNS 服务器程序每接收到一个查询或者主机 DNS 回答，创建一个新线程处理数据，防止等待数据造成的 CPU 闲置。若需要转发查询，将消息 ID 改为线程对应序号，并保存原 ID，当收到主机 DNS 消息时，依据 ID 找到对应线程，修改为原 ID 后发送消息给客户端并释放线程资源。

3、超时处理：设置较长的超时时限，若程序超时未接收到主机 DNS 信息，直接释放线程资源，不对客户端做任何回应，将超时透明传递给客户端；若程序在超时后收到主机 DNS 信息且消息 ID 对应的线程已被另一个客户端查询占用，发送回答给该客户端，该客户端依据错误 ID 丢弃回答。

4、缓存功能：采用自定义红黑树实现快速查询实现时间复杂度为 $O(\log n)$ 的增删查改，采用自定义双向链表实现 LRU 机制。程序读入“域名-IP 地址”对照表时，将该表项设置为静态数据加入红黑树。当客户端查询的域名需要转发给主机 DNS 服务器时，收到主机 DNS 回答后，将回答中的 A 和 AAAA 类型数据设置为动态数据后加入红黑树，并插入双向链表表头，若链表大小超过指定容量，删除表尾数据。若客户端查询的域名在红黑树中有对应表数据，返回对应数据，若为动态数据，将该数据插入双向链表表头。

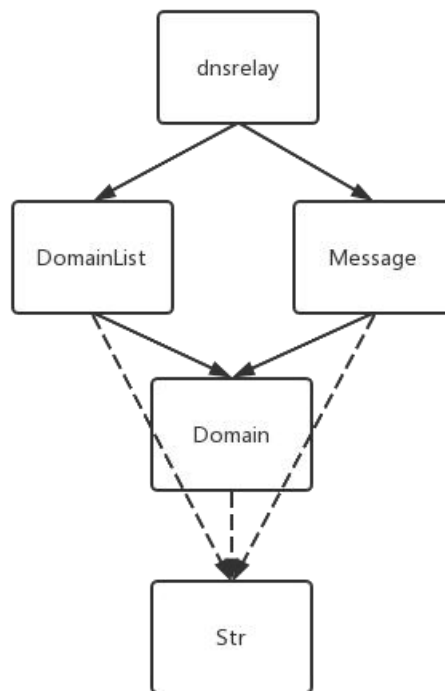
由于可能存在域名更换，域名不再使用的情况，动态缓存会定时清空。

允许一个域名保存多个 A 或 AAAA 类型 IP 地址，对于 MX、CNAME 等其他类型的查询不做缓存只进行转发。

动态缓存只存于内存中，不保存到存有静态数据的文件，避免污染静态数据。

二、模块划分和软件流程图

2.1 模块划分



2.1.1 dnsrelay

与客户端和主机 DNS 进行网络通信，实现 DNS 服务器基本功能。

1、结构体设计

```
//与主机 DNS 通信线程结构体
struct Server {
    int isOccupied;//判断是否被占用
    Str response;//接受的数据
    pthread_t tid;// 线程 ID
};

//与客户端通信线程结构体
struct Client {
    short id;//clientID, 用于标识不同客户
    short originId;//query 原 id
    int isOccupied;//判断是否被占用
    Str query;//接受的数据
```

```
pthread_t tid;// 线程 ID
SOCKADDR_IN clientAdd;// 地址信息
};
```

2、常数设计

```
#define VERSION 1.70 //程序版本号
#define MESSAGE_MAX_LENGTH 512 //报文最大长度
#define MAX_PTHREAD 100 //线程最大数量
#define SLEEP_INTERVAL 100//周期性判断是否超时间隔
#define OVERTIME 3000 //超时应答时间
#define CACHE_CAPACITY 1000//动态 cache 容量
#define REFRESH_TIME 3600000000//刷新 cache 间隔
#define OCCUPIED 1 //线程结构体被占用
#define UNOCCUPIED 0 //线程结构体未被占用
```

3、全局变量

```
struct Client clients[MAX_PTHREAD / 2];//与客户端通信线程
struct Server servers[MAX_PTHREAD / 2];//与主机 DNS 服务器通信线程
DomainList domainList = { NULL,0 };//域名-IP 表
SOCKET clientSocket;//与客户端通信 socket
SOCKET serverSocket;//与主机 DNS 服务器通信 socket
SOCKADDR_IN dnsAdd;//主机 DNS 地址信息
SOCKADDR_IN localAdd;//本地 DNS 地址信息
pthread_mutex_t output;//输出线程锁
pthread_mutex_t modifyTotalQueryNum;//修改 queryNum 线程锁
pthread_mutex_t modifyCache;//修改域名-IP 表缓存线程锁

int debugLevel = 1;//debug 等级
char* dnsIp = "218.85.157.99"; //主机 DNS 的 IP 地址
char* staticDataPath = "./dnsrelay.txt";//静态数据文件地址
int queryNum = 0;//总接受请求数量
char allowToReceive = 0;//允许接受 DNS 服务器数据
```

4、函数设计

```
void initDnsrelay(int argc, char* argv[]);
```

功能：处理命令行参数

参数：int argc - 参数个数

char* argv[] - 参数值

返回：无

```
void initSocket();
```

功能：初始化 SOCKET

参数：无

返回：无

void* cacheRefresh(void* arg);

功能：刷新动态记录

参数：无

返回：无

void* recvFromServer(void* arg);

功能：接受主机 dns 服务器数据

参数：无

返回：无

void* recvFromClient(void* arg);

功能：接受客户端数据

参数：无

返回：无

void* processResponse(void* arg);

功能：处理主机 dns 服务器数据

参数：与主机 DNS 服务器通信线程结构体

返回：无

void* processQuery(void* arg);

功能：处理客户端数据

参数：与客户端通信线程结构体

返回：无

void outputRecvInfo(const char* recvIp, const int recvPort, const Str* pOriginMessage);

功能：输出接受数据的调试信息

参数：const char* recvIp - 数据发送方 IP 地址

const int recvPort - 数据发送方端口号

const Str* pOriginMessage - 报文内容

返回：无

void outputSimpleInfo(const char* recvIp, const int isStatic, const Str* domainName, const int type);

功能：输出简单调试信息

参数：const char* recvIp - 客户端 IP 地址

const int isStatic - 是否为静态数据

const Str* domainName - 查询的域名

const int type - 查询的内容

返回：无

void outputSendInfo(const char* sendIp, const int sendPort, const Str* pNewMessage, const int oldId, const int isAddAnswer);

功能：输出发送数据的调试信息

参数：const char* sendIp - 数据接收方 IP 地址

const int sendPort - 数据接收方端口号

const Str* pNewMessage - 报文内容

const int oldId - 原报文 ID

const int isAddAnswer - 是否由本地程序添加过回答

返回：无

2.1.2 Domain

存储一个域名的 A 和 AAAA 类型 IP 地址，是否为静态数据，是否可访问。

1、结构体设计

```
typedef struct Domain {  
    Str name;//域名  
    char isStatic;//是否属于静态表  
    char isAvailable;//是否允许查询  
    char numA;//A 类型记录数量,值为-1 表示未查询  
    char num4A;//AAAA 类型记录数量,值为-1 表示未查询  
    Str* pRdataA;//A 类型记录,仅保存 RDATA 字段  
    Str* pRdata4A;//AAAA 类型记录,仅保存 RDATA 字段  
}Domain;
```

2、函数设计

```
void initDomain(Domain* pDomain);
```

功能：初始化域名

参数：Domain* pDomain - 需要初始化的域名结构体

返回：无

```
void setDomainName(Domain* pDomain, const Str* pName);
```

功能：修改域名为指定域名

参数：Domain* pDomain - 需要修改域名的域名结构体

const Str* pName - 指定域名

返回：无

```
void copyDomain(Domain* pDomain1, const Domain* pDomain2);
```

功能：复制域名结构体

参数：Domain* pDomain1 - 把修改的域名结构体

const Domain* pDomain2 - 被复制的域名结构体

返回：无

```
void addRecord(Domain* pDomain, const Str* pRdata, const int type);
```

功能：添加指定类型记录

参数：Domain* pDomain - 需要添加记录的域名结构体

const Str* pRdata - RDATA 字段记录

`const int type` - 指定类型, 1 或 28

返回: 无

`char getNumOf(const Domain* pDomain, const int type);`

功能: 获取指定类型记录的数量

参数: `const Domain* pDomain` - 需要获取记录数量的域名结构体

`const int type` - 指定类型

返回: 指定类型记录的数量

`void deleteDomain(Domain* pDomain);`

功能: 删除域名结构体

参数: `Domain* pDomain` - 需要删除的域名结构体

返回: 无

2.1.3 DomainList

存储大量域名结构体的数据结构, 采用红黑树提供 $O(\log n)$ 的增删查改, 采用自定义双向链表实现 LRU 机制。

1、结构体设计

```
typedef struct LRUCacheNode {
    Domain data;
    Str key;
    char color;
    struct LRUCacheNode* lchild;
    struct LRUCacheNode* rchild;
    struct LRUCacheNode* next;
    struct LRUCacheNode* pre;
}CacheNode, * RBTree;

typedef struct DomainList {
    struct LRUCacheNode* root;//红黑树根节点, 存储动态表和静态表, 提供快速查询
    struct LRUCacheNode head;//双向链表表头, 存储动态表, 提供 LRU 机制
    struct LRUCacheNode tail;//双向链表表尾
    int count;
    int capacity;
}DomainList;
```

2、常数设计

```
#define RED 1 //红色结点
#define BLACK 0 //黑色结点
```

3、函数设计

`void initDomainList(DomainList* pDomainList, const int capacity);`

功能: 初始化域名列表

参数: DomainList* pDomainList - 需要初始化的域名列表

const int capacity - LRU 双向链表容量

返回: 无

int addDomain(DomainList* pDomainList, const Domain* pDomain);

功能: 添加域名结构体

参数: DomainList* pDomainList - 需要加入域名结构体的域名列表

const Domain* pDomain - 需要加入的域名结构体

返回: 加入成功返回 1, 否则返回 0

void deleteDomainList(DomainList* pDomainList);

功能: 释放域名表

参数: DomainList* pDomainList - 需要被释放的域名列表

返回: 无

void refreshDomianList(DomainList* pDomainList);

功能: 刷新域名表中动态记录

参数: DomainList* pDomainList - 需要被刷新的域名列表

返回: 无

Domain* findDomain(DomainList* pDomainList, const Str* pKey);

功能: 查找指定域名的域名结构体

参数: DomainList* pDomainList - 需要被查找的域名列表

const Str* pKey - 指定域名

返回: 若找到返回域域名结构体指针, 否则返回 NULL

void loadDomainList(DomainList* pDomainList, const char* fileAddress, int debugLevel);

功能: 从文件中读取域名数据

参数: DomainList* pDomainList - 需要添加域名数据的域名列表

const char* fileAddress - 文件地址

const int debugLevel - debug 等级

返回: 无

2.1.4 Message

含有修改修改、读取报文内容的相关函数

1、函数设计

void setId(Str* pQuery, const short id);

功能: 修改 ID 字段

参数: Str* pQuery - 需要被修改的报文

const short id - 指定 id

返回: 无


```
void setQr(Str* pQuery, const char qr);
```

功能: 修改 QR 字段

参数: Str* pQuery - 需要被修改的报文
const short qr - 指定 qr

返回: 无

```
void setRcode(Str* pQuery, const char rcode);
```

功能: 修改 RCODE 字段

参数: Str* pQuery - 需要被修改的报文
const short rcode - 指定 rcode

返回: 无

```
void setTc(Str* pQuery, const char rcode);
```

功能: 修改 TC 字段

参数: Str* pQuery - 需要被修改的报文
const short tc - 指定 tc

返回: 无

```
void setAncount(Str* pQuery, const short ancount);
```

功能: 修改 ANCOUNT 字段

参数: Str* pQuery - 需要被修改的报文
const short ancount - 指定 ancount

返回: 无

```
short getId(const Str* pMessage);
```

功能: 读取 ID 字段

参数: const Str* pMessage - 需要被读取的报文

返回: 报文 ID 字段

```
int getQr(const Str* pMessage);
```

功能: 读取 QR 字段

参数: const Str* pMessage - 需要被读取的报文

返回: 报文 QR 字段

```
int getOpcode(const Str* pMessage);
```

功能: 读取 OPCODE 字段

参数: const Str* pMessage - 需要被读取的报文

返回: 报文 OPCODE 字段

```
int getAa(const Str* pMessage);
```

功能: 读取 AA 字段

参数: const Str* pMessage - 需要被读取的报文

返回: 报文 AA 字段

```
int getTc(const Str* pMessage);
```

功能：读取 TC 字段

参数：const Str* pMessage - 需要被读取的报文

返回：报文 TC 字段

```
int getRd(const Str* pMessage);
```

功能：读取 RD 字段

参数：const Str* pMessage - 需要被读取的报文

返回：报文 RD 字段

```
int getRa(const Str* pMessage);
```

功能：读取 RA 字段

参数：const Str* pMessage - 需要被读取的报文

返回：报文 RA 字段

```
int getRcode(const Str* pMessage);
```

功能：读取 RCODE 字段

参数：const Str* pMessage - 需要被读取的报文

返回：报文 RCODE 字段

```
short getQdcount(const Str* pMessage);
```

功能：读取 QDCOUNT 字段

参数：const Str* pMessage - 需要被读取的报文

返回：报文 QDCOUNT 字段

```
short getAncount(const Str* pMessage);
```

功能：读取 ANCOUNT 字段

参数：const Str* pMessage - 需要被读取的报文

返回：报文 ANCOUNT 字段

```
short getNscount(const Str* pMessage);
```

功能：读取 NSCOUNT 字段

参数：const Str* pMessage - 需要被读取的报文

返回：报文 NSCOUNT 字段

```
short getArcount(const Str* pMessage);
```

功能：读取 ARCOUNT 字段

参数：const Str* pMessage - 需要被读取的报文

返回：报文 ARCOUNT 字段

```
short getQnameAndType(Str* pName, const Str* pMessage);
```

功能：读取 QNAME 字段和 QTYPE 字段

参数：Str* pName - 需要被修改的 qname 字段

const Str* pMessage - 需要被读取的报文
返回: 若格式错误返回 0, 否则返回查询地址类型

void mergeAnswer(Str* pMessage, const Domain* pDomain, const short type);

功能: 在报文尾部连接指定类型的记录

参数: **Str* pMessage** - 需要被修改的报文

const Domain* pDomain - 含有所需域名记录的域名结构体

const short type - 指定类型

返回: 无

int saveRdata(Domain* pDomain, const Str* pMessage);

功能: 保存报文中 RDATA 记录到域名结构体,

参数: **Domain* pDomain** - 需要被修改的域名结构体

const Str* pMessage - 报文

返回: 若记录类型为 A 或 AAAA 类型加入域名并返回 1, 否则返回 0

2.1.5 Str

自定义字符串数据结构。

1、结构体设计

```
typedef struct Str {  
    char* string; //字符串, 不以 '\0' 作为结尾  
    unsigned short length; //字符串长度  
}Str;
```

2、函数设计

void initStr(Str* str);

功能: 初始化字符串

参数: **Str* str** - 需要初始化的字符串

返回: 无

void setStr(Str* dst, const char* src, const unsigned short length);

功能: 设置字符串为指定字符序列

参数: **Str* dst** - 需要修改的字符串

const char* src - 指定字符序列

const unsigned short length - 字符串长度

返回: 无

int strLength(const Str* str);

功能: 获取指定字符串的长度

参数: **const Str* str** - 指定字符串

返回: 字符串的长度

void getSubstring(Str* dst, const Str* src, const unsigned short add, const

`unsigned short length);`

功能: 读取指定字符串的子串

参数: `Str* dst` - 需要被修改的字符串

`const Str* src` - 指定字符串

`const unsigned short add` - 子串起始位置

`const unsigned short length` - 子串长度

返回: 无

`void copyStr(Str* dst, const Str* src);`

功能: 复制指定字符串

参数: `Str* dst` - 需要修改的字符串

`const Str* src` - 指定字符串

返回: 无

`void copynStr(Str* dst, const Str* src, const unsigned short length);`

功能: 复制指定字符串的指定长度

参数: `Str* dst` - 需要被修改的字符串

`const Str* src` - 指定字符串

`const unsigned short length` - 指定长度

返回: 无

`int compareStr(const Str* str1, const Str* str2);`

功能: 比较字符串

参数: `const Str* str1` - 第一个字符串

`const Str* str2` - 第二个字符串

返回: `str1` 较大返回 1, `str1` 较小返回 -1, 相等返回 0

`void concatStr(Str* dst, const Str* src);`

功能: 连接指定字符串到另一字符串尾部

参数: `Str* dst` - 需要连接的字符串

`const Str* src` - 指定字符串

返回: 无

`void appendStr(Str* dst, const char ch);`

功能: 在指定字符串尾部添加指定字符

参数: `Str* dst` - 指定字符串

`const char ch` - 指定字符

返回: 无

`void concatnStr(Str* dst, const Str* src, const unsigned short length);`

功能: 连接指定长度字符串到另一字符串尾部

参数: `Str* dst` - 需要被修改的字符串

`const Str* src` - 指定字符串

`const unsigned short length` - 指定长度

返回：无

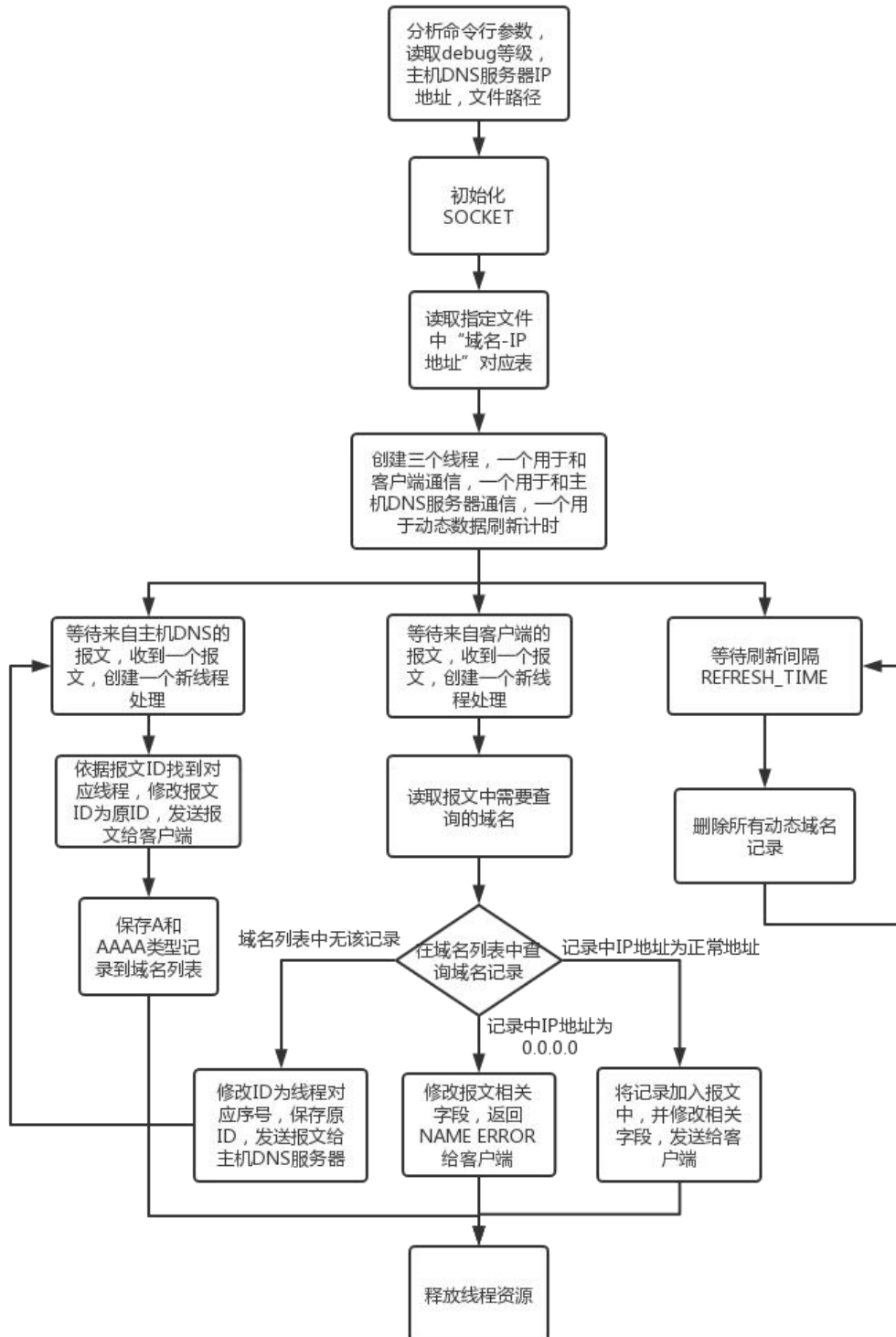
void deleteStr(Str* dst);

功能：删除字符串

参数：**Str* str** - 需要删除的字符串

返回：无

2.2 软件流程图



三、测试用例以及运行结果

以-dd 模式打开程序

```
C:\Windows\System32\cmd.exe - dnsrelay -dd
Microsoft Windows [版本 10.0.22000.739]
(c) Microsoft Corporation. 保留所有权利。

D:\dnsrelay\x64\Release>dnsrelay -dd
DNSRELAY, Version 1.70, Build: Jul 10 2022 20:46:31
Usage: dnsrelay [-d | -dd] [<dns-server>] [<db-file>]

Name server 218.85.157.99:53.
Debug level 2.
Bind UDP port 53...OK
Try to load table ".//dnsrelay.txt"...OK
0: 0.0.0.0      008.cn
1: 0.0.0.0      2qq.cn
2: 0.0.0.0      555.265.com
3: 0.0.0.0      abc.265.com
4: 0.0.0.0      abcdesign.ru
5: 0.0.0.0      ad.qingyule.com
6: 0.0.0.0      alexey.pioneers.com.ru
7: 0.0.0.0      baltnet.ru
8: 0.0.0.0      cctv1.net
9: 0.0.0.0      cctv8.net
10: 0.0.0.0     chinabdkx.363.net
11: 0.0.0.0     ciachoo.pl
12: 0.0.0.0     clients.babylon.co.il
13: 0.0.0.0     dicto.ru
14: 0.0.0.0     elemental.ru
15: 0.0.0.0     errorguard.com
16: 0.0.0.0     financial.washingtonpost.com
17: 0.0.0.0     free.bestialityhost.com
18: 0.0.0.0     friendlygreeting.com
19: 0.0.0.0     gamma.vyborg.ru
20: 0.0.0.0     gin.ru
21: 0.0.0.0     girlchinese.com
22: 0.0.0.0     glass-master.ru
23: 0.0.0.0     gutemine.wu-wien.ac.at
24: 0.0.0.0     hack-gegen-rechts.com
25: 0.0.0.0     hgrstrailer.com
26: 0.0.0.0     home.kimo.com.tw
27: 0.0.0.0     home.profootball.ru
28: 0.0.0.0     hotbar.com
29: 0.0.0.0     hothack.home.chinaren.com
30: 0.0.0.0     intellect.lvc
31: 0.0.0.0     interfoodtd.ru
32: 0.0.0.0     it.trendmicro-europe.com
33: 0.0.0.0     jewishgen.org
34: 0.0.0.0     k2kapital.com
35: 0.0.0.0     lars-s.privat.t-online.de
36: 0.0.0.0     laugh-mail.net
37: 0.0.0.0     marketscore.com
38: 0.0.0.0     math.kobe-u.ac.jp
39: 0.0.0.0     me.5e163.com
40: 0.0.0.0     momentum.ru
41: 0.0.0.0     movie.sx.zj.cn
42: 0.0.0.0     nefkom.net
43: 0.0.0.0     newyouth.3322.net
44: 0.0.0.0     no-abi2003.de
```

程序成功读入默认路径文件内的“域名-IP”对应表内 909 条数据

```
C:\Windows\System32\cmd.exe - dnsrelay -dd
873: 210.242.125.98 plus.url.google.co.jp
874: 210.242.125.98 scholar.google.co.jp
875: 210.242.125.98 toolbar.google.co.jp
876: 210.242.125.98 toolbarqueries.google.co.jp
877: 210.242.125.98 translate.google.co.jp
878: 210.242.125.98 www.google.co.jp
879: 210.242.125.98 android.googleusercontent.com
880: 210.242.125.98 auth.keyhole.com
881: 210.242.125.98 chrome.angrybirds.com
882: 210.242.125.98 developer.chrome.com
883: 210.242.125.98 domains.googleusercontent.com
884: 210.242.125.98 earthengine.googleusercontent.com
885: 210.242.125.98 g.co
886: 210.242.125.98 gmail.com
887: 210.242.125.98 goo.gl
888: 210.242.125.98 listen.googleusercontent.com
889: 210.242.125.98 m.gmail.com
890: 210.242.125.98 m.googleusercontent.com
891: 210.242.125.98 ngrams.googleusercontent.com
892: 210.242.125.98 ssl.google-analytics.com
893: 210.242.125.98 www.gmail.com
894: 210.242.125.98 www.googleadservices.com
895: 210.242.125.98 www.google-analytics.com
896: 210.242.125.98 www.googleusercontent.com
897: 210.242.125.98 www.googleusercontent.com
898: 210.242.125.98 chatenable.mail.google.com
899: 210.242.125.98 panoramio.com
900: 210.242.125.98 www.panoramio.com
901: 210.242.125.98 static.panoramio.com
902: 74.125.207.191 www.blogger.com
903: 74.125.207.132 googleblog.blogspot.com
904: 74.125.207.132 googlechinablog.blogspot.com
905: 74.125.207.132 1.bp.blogspot.com
906: 74.125.207.132 2.bp.blogspot.com
907: 74.125.207.132 3.bp.blogspot.com
908: 74.125.207.132 4.bp.blogspot.com
Successfully load 909 records
```

1、运行：nslookup - 127.0.0.1，查询 DNS 文件中 IP 地址为 0.0.0.0 的域名 008.cn，程序返回 NAME ERROR 给客户端

```
> 008.cn
服务器: localhost
Address: 127.0.0.1

*** localhost 找不到 008.cn: Non-existent domain
>
```

```
3:* 2022-07-10 22:19:42 Client 127.0.0.1 008.cn, TYPE 1, CLASS 1
SEND to 127.0.0.1:56240 (24 bytes) 00 04 85 83 00 01 00 00 00 00 00 00 03 30 30 38 02 63 6e 00 00 01 00 01
ID 0004, QR 1, OPCODE 0, AA 1, TC 0, RD 1, RA 1, Z 0, RCODE 3
QDCOUNT 1, ANCOUNT 0, NSCOUNT 0, ARCOUNT 0
RCV from 127.0.0.1:56241 (24 bytes) 00 05 01 00 00 01 00 00 00 00 00 00 03 30 30 38 02 63 6e 00 00 1c 00 01
ID 0005, QR 0, OPCODE 0, AA 0, TC 0, RD 1, RA 0, Z 0, RCODE 0
QDCOUNT 1, ANCOUNT 0, NSCOUNT 0, ARCOUNT 0
4: 2022-07-10 22:19:42 Client 127.0.0.1 008.cn, TYPE 28, CLASS 1
SEND to 202.106.0.20:53 (24 bytes) [ID 0005->df61]
RCV from 202.106.0.20:53 (24 bytes) df 61 81 80 00 01 00 00 00 00 00 00 03 30 30 38 02 63 6e 00 00 1c 00 01
ID df61, QR 1, OPCODE 0, AA 0, TC 0, RD 1, RA 1, Z 0, RCODE 0
QDCOUNT 1, ANCOUNT 0, NSCOUNT 0, ARCOUNT 0
SEND to 127.0.0.1:56241 (24 bytes) [ID df61 -> 0005]
```

2、查询 DNS 文件中 IP 地址为 123.127.134.10 的域名 www.bupt.cn，

其中域名 www.bupt.cn 的 A 类型查询在域名列表中查询到对应 IP 地址，直接返回对应的 IP 地址；

其中 AAAA 类型查询以及域名 www.bupt.cn.home 的查询，由于域名列表中无该记录且程序并未有查询过该记录，修改 ID 后转发给主机 DNS 服务器，从主机 DNS 服务器收到回答后还原 ID 发送给客户端并缓存记录。

```
*** UnKnown 找不到 008.cn: Non-existent domain
> www.bupt.cn
服务器: UnKnown
Address: 127.0.0.1

非权威应答:
名称: vn.bupt.edu.cn
Addresses: 2001:da8:215:4038::161
123.127.134.10
Aliases: www.bupt.cn
```

```
5: 2022-07-10 22:24:47 Client 127.0.0.1 www.bupt.cn.Home, TYPE 1
SEND to 218.85.157.99:53 (34 bytes) [ID 0006->0000]
RECV from 218.85.157.99:53 (109 bytes) 00 00 85 83 00 01 00 00 01 00 00 03 77 77 77 04 62 75 70 74 02 63 6e 04 48 6f
6d 65 00 00 01 00 01 00 00 06 00 01 00 00 07 08 00 40 01 61 0c 72 6f 6f 74 2d 73 65 72 76 65 72 73 03 6e 65 74 00 05 6e
73 74 6c 64 0c 76 65 72 69 73 69 67 6e 2d 67 72 73 03 63 6f 6d 00 78 86 5a d9 00 00 07 08 00 00 03 84 00 09 3a 80 00 01
51 80
ID 0000, QR 1, OPCODE 0, AA 0, TC 0, RD 1, RA 1, Z 0, RCODE 3
QDCOUNT 1, ANCOUNT 0, NSCOUNT 1, ARCOUNT 0
SEND to 127.0.0.1:51261 (109 bytes) [ID 0000->0006]
RECV from 127.0.0.1:51262 (34 bytes) 00 07 01 00 00 01 00 00 00 00 00 03 77 77 77 04 62 75 70 74 02 63 6e 04 48 6f 6
d 65 00 00 1c 00 01
ID 0007, QR 0, OPCODE 0, AA 0, TC 0, RD 1, RA 0, Z 0, RCODE 0
QDCOUNT 1, ANCOUNT 0, NSCOUNT 0, ARCOUNT 0
6: 2022-07-10 22:24:47 Client 127.0.0.1 www.bupt.cn.Home, TYPE 28
SEND to 218.85.157.99:53 (34 bytes) [ID 0007->0000]
RECV from 218.85.157.99:53 (109 bytes) 00 00 85 83 00 01 00 00 01 00 00 03 77 77 77 04 62 75 70 74 02 63 6e 04 48 6f 6f
6d 65 00 00 1c 00 01 00 00 06 00 01 00 00 07 08 00 40 01 61 0c 72 6f 6f 74 2d 73 65 72 76 65 72 73 03 6e 65 74 00 05 6e
73 74 6c 64 0c 76 65 72 69 73 69 67 6e 2d 67 72 73 03 63 6f 6d 00 78 86 5a d9 00 00 07 08 00 00 03 84 00 09 3a 80 00 01
51 80
ID 0000, QR 1, OPCODE 0, AA 0, TC 0, RD 1, RA 1, Z 0, RCODE 3
QDCOUNT 1, ANCOUNT 0, NSCOUNT 1, ARCOUNT 0
SEND to 127.0.0.1:51262 (109 bytes) [ID 0000->0007]
RECV from 127.0.0.1:51263 (29 bytes) 00 08 01 00 00 01 00 00 00 00 00 03 77 77 77 04 62 75 70 74 02 63 6e 00 00 01 0
0 01
ID 0008, QR 0, OPCODE 0, AA 0, TC 0, RD 1, RA 0, Z 0, RCODE 0
QDCOUNT 1, ANCOUNT 0, NSCOUNT 0, ARCOUNT 0
7:* 2022-07-10 22:24:47 Client 127.0.0.1 www.bupt.cn, TYPE 1
SEND to 127.0.0.1:51263 (56 bytes) 00 08 81 00 00 01 00 01 00 00 00 03 77 77 77 04 62 75 70 74 02 63 6e 00 00 01 00
01 03 77 77 77 04 62 75 70 74 02 63 6e 00 00 01 00 01 00 00 27 10 00 04 7b 7f 86 0a
ID 0008, QR 1, OPCODE 0, AA 0, TC 0, RD 1, RA 0, Z 0, RCODE 0
QDCOUNT 1, ANCOUNT 1, NSCOUNT 0, ARCOUNT 0
RECV from 127.0.0.1:51264 (29 bytes) 00 09 01 00 00 01 00 00 00 00 00 03 77 77 77 04 62 75 70 74 02 63 6e 00 00 1c 0
0 01
ID 0009, QR 0, OPCODE 0, AA 0, TC 0, RD 1, RA 0, Z 0, RCODE 0
QDCOUNT 1, ANCOUNT 0, NSCOUNT 0, ARCOUNT 0
8: 2022-07-10 22:24:47 Client 127.0.0.1 www.bupt.cn, TYPE 28
SEND to 218.85.157.99:53 (29 bytes) [ID 0009->0000]
RECV from 218.85.157.99:53 (198 bytes) 00 00 81 80 00 01 00 02 00 02 00 03 03 77 77 77 04 62 75 70 74 02 63 6e 00 00 1c
00 01 c0 0c 00 05 00 01 00 00 03 e8 00 0e 02 76 6e 04 62 75 70 74 03 65 64 75 c0 15 c0 29 00 1c 00 01 00 00 02 58 00 10
20 01 0d a8 02 15 40 38 00 00 00 00 00 01 61 c0 2c 00 02 00 01 00 00 03 74 00 0d 02 6e 73 07 62 75 70 74 6e 65 74 c0
31 c0 2c 00 02 00 01 00 00 03 74 00 06 03 67 75 73 c0 62 c0 78 00 01 00 01 00 01 f8 c9 00 04 ca 70 0a 24 c0 5f 00 1c 00
01 00 01 eb a7 00 10 20 01 0d a8 02 02 00 10 00 00 00 00 00 00 37 c0 78 00 1c 00 01 00 01 f8 c9 00 10 20 01 0d a8 02
02 00 10 00 00 00 00 00 00 36
ID 0000, QR 1, OPCODE 0, AA 0, TC 0, RD 1, RA 1, Z 0, RCODE 0
QDCOUNT 1, ANCOUNT 2, NSCOUNT 3, ARCOUNT 3
SEND to 127.0.0.1:51264 (198 bytes) [ID 0000->0009]
```

3、查询 DNS 文件中没有的域名 www.bilibili.com

由于域名列表中无该记录且程序并未有查询过该记录，修改 ID 后转发给主机 DNS 服务器，从主机 DNS 服务器收到回答后还原 ID 发送给客户端并缓存记录。

非权威应答:
名称: vn.bupt.edu.cn
Addresses: 2001:da8:215:4038::161
123.127.134.10
Aliases: www.bupt.cn

> www.bilibili.com
服务器: UnKnown
Address: 127.0.0.1

非权威应答:
名称: a.w.bilibidn1.com
Addresses: 240e:97d:2000:a06::1
240e:97d:2000:a06::4
240e:97d:2000:a06::2
240e:97d:2000:a06::5
240e:97d:2000:a06::3
59.36.232.5
59.36.232.2
59.36.232.4
59.36.232.3
59.36.232.1
Aliases: www.bilibili.com

```
RECV from 127.0.0.1:52912 (39 bytes) 00 0a 01 00 00 01 00 00 00 00 00 00 03 77 77 77 08 62 69 6c 69 62 69 6c 69 03 63 6
f 6d 04 48 6f 6d 65 00 00 01 00 01
ID 000a, QR 0, OPCODE 0, AA 0, TC 0, RD 1, RA 0, Z 0, RCODE 0
QDCOUNT 1, ANCOUNT 0, NSCOUNT 0, ARCOUNT 0
9: 2022-07-10 22:33:18 Client 127.0.0.1 www.bilibili.com Home, TYPE 1
SEND to 218.85.157.99:53 (39 bytes) [ID 000a->0000]
RECV from 218.85.157.99:53 (114 bytes) 00 00 85 83 00 01 00 00 00 01 00 00 03 77 77 77 08 62 69 6c 69 62 69 6c 69 03 63
6f 6d 04 48 6f 6d 65 00 00 01 00 01 00 00 06 00 01 00 00 07 08 00 40 01 61 0c 72 6f 6f 74 2d 73 65 72 76 65 72 73 03 6e
65 74 00 05 6e 73 74 6c 64 0c 76 65 72 69 73 69 67 6e 2d 67 72 73 03 63 6f 6d 00 78 86 5a d9 00 00 07 08 00 00 03 84 00
09 3a 80 00 01 51 80
ID 0000, QR 1, OPCODE 0, AA 0, TC 0, RD 1, RA 1, Z 0, RCODE 3
QDCOUNT 1, ANCOUNT 0, NSCOUNT 1, ARCOUNT 0
SEND to 127.0.0.1:52912 (114 bytes) [ID 0000->000a]
RECV from 127.0.0.1:52913 (39 bytes) 00 0b 01 00 00 01 00 00 00 00 00 00 03 77 77 77 08 62 69 6c 69 62 69 6c 69 03 63 6
f 6d 04 48 6f 6d 65 00 00 1c 00 01
ID 000b, QR 0, OPCODE 0, AA 0, TC 0, RD 1, RA 0, Z 0, RCODE 0
QDCOUNT 1, ANCOUNT 0, NSCOUNT 0, ARCOUNT 0
10: 2022-07-10 22:33:18 Client 127.0.0.1 www.bilibili.com Home, TYPE 28
SEND to 218.85.157.99:53 (39 bytes) [ID 000b->0000]
RECV from 218.85.157.99:53 (114 bytes) 00 00 85 83 00 01 00 00 00 01 00 00 03 77 77 77 08 62 69 6c 69 62 69 6c 69 03 63
6f 6d 04 48 6f 6d 65 00 00 1c 00 01 00 00 06 00 01 00 00 07 08 00 40 01 61 0c 72 6f 6f 74 2d 73 65 72 76 65 72 73 03 6e
65 74 00 05 6e 73 74 6c 64 0c 76 65 72 69 73 69 67 6e 2d 67 72 73 03 63 6f 6d 00 78 86 5a d9 00 00 07 08 00 00 03 84 00
09 3a 80 00 01 51 80
ID 0000, QR 1, OPCODE 0, AA 0, TC 0, RD 1, RA 1, Z 0, RCODE 3
QDCOUNT 1, ANCOUNT 0, NSCOUNT 1, ARCOUNT 0
SEND to 127.0.0.1:52913 (114 bytes) [ID 0000->000b]
RECV from 127.0.0.1:52914 (34 bytes) 00 0c 01 00 00 01 00 00 00 00 00 00 03 77 77 77 08 62 69 6c 69 62 69 6c 69 03 63 6
f 6d 00 00 01 00 01
ID 000c, QR 0, OPCODE 0, AA 0, TC 0, RD 1, RA 0, Z 0, RCODE 0
QDCOUNT 1, ANCOUNT 0, NSCOUNT 0, ARCOUNT 0
11: 2022-07-10 22:33:18 Client 127.0.0.1 www.bilibili.com, TYPE 1
SEND to 218.85.157.99:53 (34 bytes) [ID 000c->0000]
RECV from 218.85.157.99:53 (463 bytes) 00 00 81 80 00 01 00 06 00 02 00 10 03 77 77 77 08 62 69 6c 69 62 69 6c 69 03 63
6f 6d 00 00 01 00 01 c0 0c 00 05 00 01 00 00 02 58 00 0f 01 61 01 77 08 62 69 6c 69 63 64 6e 31 c0 19 c0 2e 00 01 00 01
00 00 02 58 00 04 3b 24 e8 05 c0 2e 00 01 00 01 00 00 02 58 00 04 3b 24 e8 02 c0 2e 00 01 00 01 00 00 02 58 00 04 3b 24
e8 04 c0 2e 00 01 00 01 00 00 02 58 00 04 3b 24 e8 03 c0 2e 00 01 00 01 00 00 02 58 00 04 3b 24 e8 01 c0 32 00 02 00 01
00 00 05 03 00 0c 03 6e 73 33 05 64 6e 73 76 35 c0 19 c0 32 00 02 00 01 00 00 05 03 00 06 03 6e 73 34 c0 9d c0 b1 00 1c
00 01 00 01 56 ad 00 10 24 02 4e 00 10 20 12 64 00 00 91 36 29 b6 fc 32 c0 99 00 1c 00 01 00 01 16 81 00 10 24 02 4e 00
14 30 11 02 00 00 91 36 2b ba 61 c0 b1 00 01 00 01 00 01 51 4f 00 04 6a 37 52 4c c0 b1 00 01 00 01 51 4f 00 04
a2 0e 18 f8 c0 b1 00 01 00 01 00 01 51 4f 00 04 a2 0e 19 f8 c0 b1 00 01 00 01 51 4f 00 04 b7 c0 a4 77 c0 b1 00 01
00 01 00 01 51 4f 00 04 df a6 97 7e c0 b1 00 01 00 01 00 01 51 4f 00 04 3b 24 78 93 c0 b1 00 01 00 01 51 4f 00 04
65 e2 dc 0c c0 99 00 01 00 01 00 01 4e 87 00 04 a2 0e 18 fb c0 99 00 01 00 01 00 01 4e 87 00 04 a2 0e 19 fb c0 99 00 01
00 01 00 01 4e 87 00 04 b7 c0 c9 5e c0 99 00 01 00 01 00 01 4e 87 00 04 df a6 97 10 c0 99 00 01 00 01 00 01 4e 87 00 04
31 ea af 67 c0 99 00 01 00 01 00 01 4e 87 00 04 3d 97 b4 33 c0 99 00 01 00 01 00 01 4e 87 00 04 a2 0e 12 bc
ID 0000, QR 1, OPCODE 0, AA 0, TC 0, RD 1, RA 1, Z 0, RCODE 0
QDCOUNT 1, ANCOUNT 6, NSCOUNT 2, ARCOUNT 16
SEND to 127.0.0.1:52914 (463 bytes) [ID 0000->000c]
RECV from 127.0.0.1:52915 (34 bytes) 00 0d 01 00 00 01 00 00 00 00 00 00 03 77 77 77 08 62 69 6c 69 62 69 6c 69 03 63 6
f 6d 00 00 1c 00 01
ID 000d, QR 0, OPCODE 0, AA 0, TC 0, RD 1, RA 0, Z 0, RCODE 0
QDCOUNT 1, ANCOUNT 0, NSCOUNT 0, ARCOUNT 0
12: 2022-07-10 22:33:18 Client 127.0.0.1 www.bilibili.com, TYPE 28
SEND to 218.85.157.99:53 (34 bytes) [ID 000d->0000]
RECV from 218.85.157.99:53 (495 bytes) 00 00 81 80 00 01 00 06 00 02 00 0f 03 77 77 77 08 62 69 6c 69 62 69 6c 69 03 63
6f 6d 00 00 1c 00 01 c0 0c 00 05 00 01 00 00 02 58 00 0f 01 61 01 77 08 62 69 6c 69 63 64 6e 31 c0 19 c0 2e 00 1c 00 01
00 00 02 58 00 10 24 0e 09 7d 20 00 0a 06 00 00 00 00 00 00 00 00 01 c0 2e 00 1c 00 01 00 00 02 58 00 10 24 0e 09 7d 20 00
0a 06 00 00 00 00 00 00 04 c0 2e 00 1c 00 01 00 00 02 58 00 10 24 0e 09 7d 20 00 0a 06 00 00 00 00 00 00 02 c0 2e
00 1c 00 01 00 00 02 58 00 10 24 0e 09 7d 20 00 0a 06 00 00 00 00 00 00 00 00 05 c0 2e 00 1c 00 01 00 00 02 58 00 10 24 0e
09 7d 20 00 0a 06 00 00 00 00 00 00 00 03 c0 32 00 02 00 01 00 00 06 0b 0c 03 6e 73 33 05 64 6e 73 76 35 c0 19 c0 32
00 02 00 01 00 00 06 0b 00 06 03 6e 73 34 c0 d9 c0 ed 00 01 00 01 00 01 52 57 00 04 3b 24 78 93 c0 ed 00 01 00 01 00 01
52 57 00 04 65 e2 dc 0c c0 ed 00 01 00 01 00 01 52 57 00 04 6a 37 52 4c c0 ed 00 01 00 01 00 01 52 57 00 04 a2 0e 18 f8
c0 ed 00 01 00 01 00 01 52 57 00 04 a2 0e 19 f8 c0 ed 00 01 00 01 00 01 52 57 00 04 b7 c0 a4 77 c0 ed 00 01 00 01 00 01
52 57 00 04 df a6 97 7e c0 d5 00 01 00 01 00 01 4f 8f 00 04 3d 97 b4 33 c0 d5 00 01 00 01 00 01 4f 8f 00 04 a2 0e 12 bc
c0 d5 00 01 00 01 00 01 4f 8f 00 04 a2 0e 18 fb c0 d5 00 01 00 01 00 01 4f 8f 00 04 a2 0e 19 fb c0 d5 00 01 00 01 00 01
4f 8f 00 04 b7 c0 c9 5e c0 d5 00 01 00 01 00 01 4f 8f 00 04 df a6 97 10 c0 d5 00 01 00 01 00 01 4f 8f 00 04 31 ea af 67
c0 ed 00 1c 00 01 00 01 57 b5 00 10 24 02 4e 00 10 20 12 64 00 00 91 36 29 b6 fc 32
ID 0000, QR 1, OPCODE 0, AA 0, TC 0, RD 1, RA 1, Z 0, RCODE 0
QDCOUNT 1, ANCOUNT 6, NSCOUNT 2, ARCOUNT 15
SEND to 127.0.0.1:52915 (495 bytes) [ID 0000->000d]
```

4、再次查询 DNS 文件中没有的域名 www.bilibili.com

由于之前已经缓存过该域名记录，在域名列表中查询到该域名记录，将记录直接添加到报文中，修改个字段后直接发送给客户端，不再向主机 DNS 发送查询。

```
> www.bilibili.com
服务器: UnKnown
Address: 127.0.0.1

非权威应答:
名称: www.bilibili.com
Addresses: 240e:97d:2000:a06::1
           240e:97d:2000:a06::4
           240e:97d:2000:a06::2
           240e:97d:2000:a06::5
           240e:97d:2000:a06::3
           59.36.232.5
           59.36.232.2
           59.36.232.4
           59.36.232.3
           59.36.232.1
```

```
RECV from 127.0.0.1:52964 (39 bytes) 00 0e 01 00 00 01 00 00 00 00 00 03 77 77 77 08 62 69 6c 69 62 69 6c 69 03 63 6f
6d 04 48 6f 6d 65 00 00 01 00 01
ID 000e, QR 0, OPCODE 0, AA 0, TC 0, RD 1, RA 0, Z 0, RCODE 0
QDCOUNT 1, ANCOUNT 0, NSCOUNT 0, ARCOUNT 0
13: 2022-07-10 22:35:39 Client 127.0.0.1 www.bilibili.com.Home, TYPE 1
SEND to 127.0.0.1:52964 (39 bytes) 00 0e 81 00 00 01 00 00 00 00 00 03 77 77 77 08 62 69 6c 69 62 69 6c 69 03 63 6f
6d 04 48 6f 6d 65 00 00 01 00 01
ID 000e, QR 1, OPCODE 0, AA 0, TC 0, RD 1, RA 0, Z 0, RCODE 0
QDCOUNT 1, ANCOUNT 0, NSCOUNT 0, ARCOUNT 0
RECV from 127.0.0.1:52965 (39 bytes) 00 0f 01 00 00 01 00 00 00 00 00 03 77 77 77 08 62 69 6c 69 62 69 6c 69 03 63 6f
6d 04 48 6f 6d 65 00 00 1c 00 01
ID 000f, QR 0, OPCODE 0, AA 0, TC 0, RD 1, RA 0, Z 0, RCODE 0
QDCOUNT 1, ANCOUNT 0, NSCOUNT 0, ARCOUNT 0
14: 2022-07-10 22:35:39 Client 127.0.0.1 www.bilibili.com.Home, TYPE 28
SEND to 127.0.0.1:52965 (39 bytes) 00 0f 81 00 00 01 00 00 00 00 00 03 77 77 77 08 62 69 6c 69 62 69 6c 69 03 63 6f
6d 04 48 6f 6d 65 00 00 1c 00 01
ID 000f, QR 1, OPCODE 0, AA 0, TC 0, RD 1, RA 0, Z 0, RCODE 0
QDCOUNT 1, ANCOUNT 0, NSCOUNT 0, ARCOUNT 0
RECV from 127.0.0.1:52966 (34 bytes) 00 10 01 00 00 01 00 00 00 00 00 03 77 77 77 08 62 69 6c 69 62 69 6c 69 03 63 6f
6d 00 00 01 00 01
ID 0010, QR 0, OPCODE 0, AA 0, TC 0, RD 1, RA 0, Z 0, RCODE 0
QDCOUNT 1, ANCOUNT 0, NSCOUNT 0, ARCOUNT 0
15: 2022-07-10 22:35:39 Client 127.0.0.1 www.bilibili.com, TYPE 1
SEND to 127.0.0.1:52966 (194 bytes) 00 10 81 00 00 01 00 05 00 00 00 03 77 77 77 08 62 69 6c 69 62 69 6c 69 03 63 6f
6d 00 00 01 00 01 03 77 77 77 08 62 69 6c 69 62 69 6c 69 03 63 6f 6d 00 00 01 00 01 00 00 27 10 00 04 3b 24 e8 05 03 77
77 77 08 62 69 6c 69 62 69 6c 69 03 63 6f 6d 00 00 01 00 01 00 00 27 10 00 04 3b 24 e8 02 03 77 77 77 08 62 69 6c 69 62
69 6c 69 03 63 6f 6d 00 00 01 00 01 00 00 27 10 00 04 3b 24 e8 04 03 77 77 77 08 62 69 6c 69 62 69 6c 69 03 63 6f 6d 00
00 01 00 01 00 00 27 10 00 04 3b 24 e8 03 03 77 77 77 08 62 69 6c 69 62 69 6c 69 03 63 6f 6d 00 00 01 00 01 00 00 27 10
00 04 3b 24 e8 01
ID 0010, QR 1, OPCODE 0, AA 0, TC 0, RD 1, RA 0, Z 0, RCODE 0
QDCOUNT 1, ANCOUNT 5, NSCOUNT 0, ARCOUNT 0
RECV from 127.0.0.1:52967 (34 bytes) 00 11 01 00 00 01 00 00 00 00 00 03 77 77 77 08 62 69 6c 69 62 69 6c 69 03 63 6f
6d 00 00 1c 00 01
ID 0011, QR 0, OPCODE 0, AA 0, TC 0, RD 1, RA 0, Z 0, RCODE 0
QDCOUNT 1, ANCOUNT 0, NSCOUNT 0, ARCOUNT 0
16: 2022-07-10 22:35:39 Client 127.0.0.1 www.bilibili.com, TYPE 28
SEND to 127.0.0.1:52967 (254 bytes) 00 11 81 00 00 01 00 05 00 00 00 03 77 77 77 08 62 69 6c 69 62 69 6c 69 03 63 6f
6d 00 00 1c 00 01 03 77 77 77 08 62 69 6c 69 62 69 6c 69 03 63 6f 6d 00 00 1c 00 01 00 00 27 10 00 10 24 0e 09 7d 20 00
0a 06 00 00 00 00 00 00 00 01 03 77 77 77 08 62 69 6c 69 62 69 6c 69 03 63 6f 6d 00 00 1c 00 01 00 00 27 10 00 10 24 0e
09 7d 20 00 0a 06 00 00 00 00 00 00 04 03 77 77 77 08 62 69 6c 69 62 69 6c 69 03 63 6f 6d 00 00 1c 00 01 00 00 27 10
00 10 24 0e 09 7d 20 00 0a 06 00 00 00 00 00 00 02 03 77 77 77 08 62 69 6c 69 62 69 6c 69 03 63 6f 6d 00 00 1c 00 01
00 00 27 10 00 10 24 0e 09 7d 20 00 0a 06 00 00 00 00 00 00 05 03 77 77 77 08 62 69 6c 69 62 69 6c 69 03 63 6f 6d 00
00 1c 00 01 00 00 27 10 00 10 24 0e 09 7d 20 00 0a 06 00 00 00 00 00 00 03
ID 0011, QR 1, OPCODE 0, AA 0, TC 0, RD 1, RA 0, Z 0, RCODE 0
QDCOUNT 1, ANCOUNT 5, NSCOUNT 0, ARCOUNT 0
```

四、调试中遇到并解决的问题

1、使用同一个 socket 对客户端和主机 DNS 服务器发送和接受数据，对客户端可以正常收发，对服务器发送汇报 10051 错误，解释为无法连接的网络尝试套接字操作

解决方法：在尝试了网上的多种方法无果后，最终只能选择创建两个套接字，一个用于与客户端通信，一个用于与主机 DNS 通信，成功解决问题。

2、采用两个 socket 和两个线程分别对客户端和主机 DNS 服务器进行收发报文，会导致程序收到的第一个查询报文被与主机 DNS 服务器进行收发的线程识别

解决方法：设置全局变量，当需要接受主机 DNS 服务器报文时设置为 1，与主机 DNS 服务器进行收发的线程才开始接受数据。

3、第一次采用自定义红黑树和双向链表结合的数据结构实现快速查询和 LRU 机制，程序在多次删除域名数据后报错

解决方法：经过多次调试，发现红黑树结点在左右子树都存在的情况下，需要交换需要删除的结点和其左子树中最大的结点的数据，在删除的过程中会打乱了双向链表的顺序，最终修改为直接交换两个结点双亲结点和左右子而非数据。

五、总结和心得体会

本实验从程序设计、代码编写到报告撰写均有本人一人完成。通过本实验，本人通过查询各种网上资料自学了网络通信和多线程编程的相关知识，培养了自学和查询资料能力。同时，由于全程序使用 C 语言编写，培养了本人的 C 语言程序编写能力，不过也通过本实验进一步了解了对于 C 语言的理解还仅停留在计导的水平，在阅读了许多 C++容器的底层数据结构后发现自己对于 C 语言了解和使用尚浅。最后，通过本实验本人进一步了解了 DNS 协议的实现过程以及 wireshark 软件的使用，巩固了课堂知识。