



دانشکده‌ی مهندسی کامپیوتر

پاییز ۱۳۹۶

CE-40695

شبکه‌های کامپیوتری

استاد: مهدی جعفری

در این تمرین یک DNS Server^۱ پیاده‌سازی می‌کند. این پروتکل روشی توزیع‌شده برای مربوط کردن اسم‌ها به آی‌پی‌ها است. برنامه‌ی شما باید روی پورت ۱۵۳۵۳ برای درخواست‌ها گوش کند و به هر درخواست پاسخ دهد. برای هر درخواست برنامه‌ی شما باید یک فایل متنی هم تولید بکند. جزئیات هر بخش در ادامه توضیح داده می‌شوند.

۱.۰ درخواست‌ها و پاسخ‌ها

برنامه‌ی شما باید درخواست‌های معمولی را که شامل یک آدرس هستند بپذیرد و بدون توجه به بیت RD^۲ در هدر به صورت بازگشتی آدرس آی‌پی مربوط به آن‌ها را پیدا کند.^۳ توجه کنید که این‌گونه درخواست‌ها باید قسمت‌های QR و OPCODE هر دو صفر باشند و در غیر این صورت پاسخ شما باید از نوع refused باشد که در ادامه توضیح داده می‌شود.

در ادامه برنامه‌ی شما باید به صورت بازگشتی از Root DNS Server شروع به پرسش کند. برای این کار به آی‌پی یکی از این سرورها نیاز دارید. به همین دلیل برنامه شما در هنگام شروع باید یک ورودی بگیرد که این ورودی آی‌پی همان سروری هست که استفاده می‌شود.

بعد از دریافت پاسخ از سرور ریشه باید در قسمت‌های Answer، Authority و Additional تمام رکوردهای^۴ نوع A^۵ را بررسی کند. هر کدام از این رکوردها در قسمت داده یک آی‌پی دارند که باید کوچکترین از لحاظ عددی انتخاب شود و درخواست بعدی به آن آی‌پی زده شود. به همین ترتیب کار ادامه پیدا می‌کند تا جایی که یکی از رکوردهای A در فیلد name مقداری برابر با آدرس درخواست‌شده داشته باشد. آدرس‌ها را بدون پروتکل (مثلاً http) و بدون www در اولشان در نظر بگیرید و برابری را بررسی کنید. توجه کنید در قسمت Question فیلد QTYPE را برابر با ۲۵۵ (به معنی همه‌ی نوع‌ها) و QCLASS را برابر با ۱ (به معنی اینترنت) در نظر بگیرید بدون توجه به این که درخواست چه مقادیری در این فیلدها داشت. اگر چند آدرس این‌گونه موجود بود کمترین مقدار عددی را انتخاب کنید و به عنوان جواب بفرستید.

توجه کنید که این رکورد باید در قسمت Answer قرار بگیرد. در نهایت این را به همان آدرسی که درخواست زده بود بفرستید. RCode بسته‌ی شما باید ۰ باشد. (به معنای No Error) به غیر از بیت QR که باید ۱ باشد (به معنای پاسخ)، بیت تعداد رکوردهای قسمت Answer که باز هم باید یک باشد و قسمت Message ID که باید برابر با درخواست باشد. بقیه‌ی بیت‌های Header را صفر قرار دهید.

در قسمت Answer هم name همان اسمی است که در رکورد نهایی پیدا کردید type هم از نوع A است و CLASS برابر با ۱ و ttl ۳۶۰۰ هستند. قسمت RDATA هم همان آی‌پی است.

در این نوع درخواست‌ها باید همیشه از آی‌پی برای وصل شدن به سرورها استفاده کنید و استفاده از آدرس متنی سرورها نمره‌ای

^۱ Domain Name Server

^۲ Recursion Desired

^۳ برای آشنایی بیشتر با پروتکل به RFC‌های شماره ۱۰۳۴ و ۱۰۳۵ و یا لینک <http://www.zytrax.com/books/dns/ch15/> مراجعه کنید.

^۴ RR

^۵ IPv4 Address Record

ندارد. نوع دیگری از درخواست‌ها، درخواست‌های برعکس هستند. یعنی در این درخواست‌ها آی‌پی به شما داده می‌شود و باید آدرس مربوط به آن آی‌پی را پیدا کنید. نوع معمول این درخواست‌ها به این شکل است که اگر آی‌پی به شکل a.b.c.d باشد باید به شکل d.c.b.a.in-addr.arpa در پکت باشد و شما هم در هنگام فرستاده کوثری به سرورها باید از همین فرم استفاده کنید اما در هنگام بررسی کد شما آی‌پی‌ها به طور معمولی (همان a.b.c.d) به شما داده می‌شوند و باید آن‌ها را تغییر دهید به سرور بفرستید. فیلد Opcode باز هم صفر است (هم در پکتی که به کد شما فرستاده می‌شود و هم در پکتی که شما برای سرور می‌فرستید) ۶ بقیه فیلدها را مانند قسمت قبل مشخص کنید. در این قسمت باز هم اگر جوابی از نوع A بود از آن و اگر چند تا بودند از کم‌ترین مقدار استفاده کنید ولی اگر هیچ رکوردی از نوع A نبود در بین رکوردهای NS کم‌ترین را (با در نظر گرفتن ترتیب دیکشنری) انتخاب کنید. البته این رکوردها متنی هستند و آی‌پی نیستند ولی استثنأ در این قسمت وصل شدن به اسم مانعی ندارد. در نهایت هنگامی که رکوردی از نوع PTR پیدا شد که فیلد name در آن برابر با آی‌پی آدرس درخواست شده (بعد از تبدیل فرمت) بود آن را به آدرسی که درخواست زده بود برگردانید. فیلدهای name و rdata در پاسخ را برابر با همانی که DNS Server به شما داد، قرار دهید سایر فیلدها را هم مانند بالا تعیین کنید. فقط دقت شود که این دفعه پاسخ از نوع PTR است.

۲.۰ لاگ

برای هر درخواستی که به کد شما زده می‌شود باید یک فایل متنی با فرمت txt بسازید و اسم آن را هم برابر با Message ID درخواست قرار دهید. فرمت این فایل به شکل زیر است: در هر زمانی که به یک آی‌پی وصل می‌شوید باید خط زیر را چاپ کنید:

```
connecting to [IP]
=====
```

بعد از دریافت هر پکت از طرف DNS Server ها باید آن پکت را هم در این فایل چاپ کنید که فرمت آن توضیح داده می‌شود. (این کار برای پکت که به عنوان درخواست به کد شما زده می‌شود نباید انجام شود). هر پکت دارای ۵ بخش است Header، Question، Answer، Authority و Additional. در شروع هر کدام از این قسمت‌ها باید اسم به علاوه‌ی ۱۵ علامت مساوی در فایل بنویسید. در پایان هر قسمت هم باید ۱۵ تا مساوی چاپ کنید. که به ترتیب در پایین آمده‌اند.

```
HEADER
=====
QUESTION
=====
ANSWER
=====
AUTHORITY
=====
```

توجه شود پکت‌های هر قسمت باید بعد از اسم آن قسمت و علامت‌های مساوی نوشته شوند. فرمت نوشتن پکت‌ها در ادامه توضیح داده می‌شود. برای هر قسمت باید در بین در آکولاد اسم فیلدها و مقدارشان را به ترتیب الفبایی اسم فیلدها و در هر خط یک فیلد، در فایل بنویسید.

قسمت Header باید به شکل زیر باشد: (مشخصاً مقادیر فرضی هستند و شما باید بر اساس پکت دریافتی آن‌ها را بنویسید.)

```
{
additional count : 14
answer count : 0
authority count : 13
```

۶درست‌تر این بود که هر دو یک باشند اما این حالت کار می‌کند و مد نظر است.

```

id : 38145
is authoritative : False
is response : True
is truncated : False
opcode : 0
question count : 1
recursion available : False
recursion desired : False
reserved : 0
response code : No Error
}

```

اسم فیلدها و فرمت خروجی باید دقیقا مانند بالا باشد. در ادامه سایر قسمت‌ها هم بررسی می‌شود و خروجی شما باید از لحاظ ساختار و ترتیب دقیقا مانند آن‌ها باشد. برای فیلد response code مقادیر عددی را با توجه به قالب پایین به متن تبدیل کنید.

```

0: No Error
1: Format Error
2: Server Failure
3: Name Error
4: Not Implemented
5: Refused

```

برای قسمت Question به شکل زیر عمل کنید.

```

{
Domain Name : bing.com
Query Class : 1
Query Type : 255
}

```

ساختار قسمت‌های باقی‌مانده مانند هم است پس فقط یکی از آن‌ها مثلا Authority را بررسی می‌کنیم. در هر کدام از این قسمت‌ها تعدادی پاسخ وجود دارد که هر کدام از آن‌ها می‌توانند از انواع زیر باشند:

```

A
NS
CNAME
SOA
PTR
MX
AAAA
TXT

```

همه‌ی این انواع یک ساختار مشترک به شکل زیر دارند:

```

{
class : ans_class
name : name
rdata : rdata
rdlength : rdlength
ttl : ttl
type : ans_type
}

```

برای قسمت type باید اعداد را به اسامی بالا تبدیل کنید مگر این‌که عدد جزو تایپ‌های بالا نباشد که در آن صورت عدد را در فایل بنویسید. اما قسمت rdata برای هر کدام فرق می‌کند که در ادامه توضیح داده می‌شوند. در نوع A این فیلد یک آدرس آی‌پی است که باید به شکل آی‌پی چاپ شود.

rdata : 30.83.55.192

در نوع AAAA این فیلد یک آدرس آی پی ۶ است که باید به شکل آی پی ۶ چاپ شود.

rdata : 2001:0503:a83e:0000:0000:0000:0002:0030

توجه شود جواب را مخفف نکنید و به شکل ۸ عدد ۴ رقمی در مبنای ۱۶ که با «:» جدا شده اند نشان دهید
نوع MX

```
rdata :  
{  
Mail Exchanger : mta6.am0.yahoodns.net  
Preference : 1  
}
```

نوع SOA

```
rdata :  
{  
Admin MB : hostmaster.yahoo-inc.com  
Expiration Limit : 1814400  
Minimum TTL : 600  
Primary NS : ns1.yahoo.com  
Refresh interval : 3600  
Retry interval : 300  
Serial Number : 2017101218  
}
```

انواع cname و ptr و ns مانند هم هستند و فقط شامل یک رشته هستند.

rdata : ns4.yahoo.com

نوع TXT هم شامل یک رشته است اما این رشته به شکل معمولی ذخیره شده است در سایر انواع این رشته یک encoding خاص دارد که می توانید از لینک هایی که در پانوشت صفحه ی اول هستند مطالعه کنید

rdata : #v=spf1 redirect=_spf.mail.yahoo.com

انواع دیگری از پکت های جواب هم موجود هستند که برای آن ها لازم نیست rdata را پارس کنید و بسیار کم کاربرد هستند و برای آن ها کافی است در فایل خروجی rdata را بنویسید و جلوی آن را خالی بگذارید.

rdata :

و در مقابل فیلد type هم مقدار عددی نوع آن ها را چاپ کنید. به عنوان مثال اگر از نوع WKS بود به شکل زیر عمل کنید:

type : 11

در نهایت به عنوان مثال فایل اگر ورودی به کد شما آی پی 198.41.0.4 بود و پکت درخواستی برای yahoo.com و با آی دی ۷ بود فایل شما باید با اسم 7.txt و شامل خطوط زیر باشد: (آی دی هایی که به سرور درخواست می زنند ممکن است عوض شوند و همچنین ترتیب پاسخ ها در هر گروه هم ممکن است عوض شود همچنین ممکن است جواب هایی از سرورها بگیرد که با جواب های زیر فرق داشته باشند. یا ttl و rdlength فرق کند.)

connecting to 4.0.41.198

=====

HEADER

=====

{

additional count : 12
answer count : 0
authority count : 13
id : 12108
is authoritative : False
is response : True
is truncated : False
opcode : 0
question count : 1
recursion available : False
recursion desired : False
reserved : 0
response code : No Error
}

=====

QUESTION

=====

{
Domain Name : yahoo.com
Query Class : 1
Query Type : 255
}

=====

ANSWER

=====

=====

AUTHORITY

=====

{
class : 1
name : com
rdata : e.gtld-servers.net
rdlength : 20
ttl : 172800
type : NS
}
{
class : 1
name : com
rdata : b.gtld-servers.net
rdlength : 4
ttl : 172800
type : NS
}
{
class : 1
name : com
rdata : j.gtld-servers.net
rdlength : 4
ttl : 172800
type : NS
}
{
class : 1

```
name : com
rdata : m.gtld-servers.net
rdlength : 4
ttl : 172800
type : NS
}
{
class : 1
name : com
rdata : i.gtld-servers.net
rdlength : 4
ttl : 172800
type : NS
}
{
class : 1
name : com
rdata : f.gtld-servers.net
rdlength : 4
ttl : 172800
type : NS
}
{
class : 1
name : com
rdata : a.gtld-servers.net
rdlength : 4
ttl : 172800
type : NS
}
{
class : 1
name : com
rdata : g.gtld-servers.net
rdlength : 4
ttl : 172800
type : NS
}
{
class : 1
name : com
rdata : h.gtld-servers.net
rdlength : 4
ttl : 172800
type : NS
}
{
class : 1
name : com
rdata : l.gtld-servers.net
rdlength : 4
ttl : 172800
type : NS
}
}
```

```

{
  class : 1
  name : com
  rdata : k.gtld-servers.net
  rdlength : 4
  ttl : 172800
  type : NS
}
{
  class : 1
  name : com
  rdata : c.gtld-servers.net
  rdlength : 4
  ttl : 172800
  type : NS
}
{
  class : 1
  name : com
  rdata : d.gtld-servers.net
  rdlength : 4
  ttl : 172800
  type : NS
}
=====
ADDITIONAL
=====
{
  class : 1
  name : e.gtld-servers.net
  rdata : 30.94.12.192
  rdlength : 4
  ttl : 172800
  type : A
}
{
  class : 1
  name : e.gtld-servers.net
  rdata : 2001:0502:1ca1:0000:0000:0000:0030
  rdlength : 16
  ttl : 172800
  type : AAAA
}
{
  class : 1
  name : b.gtld-servers.net
  rdata : 30.14.33.192
  rdlength : 4
  ttl : 172800
  type : A
}
{
  class : 1
  name : b.gtld-servers.net

```

```

rdata : 2001:0503:231d:0000:0000:0000:0002:0030
rdlength : 16
ttl : 172800
type : AAAA
}
{
class : 1
name : j.gtld-servers.net
rdata : 30.79.48.192
rdlength : 4
ttl : 172800
type : A
}
{
class : 1
name : j.gtld-servers.net
rdata : 2001:0502:7094:0000:0000:0000:0000:0030
rdlength : 16
ttl : 172800
type : AAAA
}
{
class : 1
name : m.gtld-servers.net
rdata : 30.83.55.192
rdlength : 4
ttl : 172800
type : A
}
{
class : 1
name : m.gtld-servers.net
rdata : 2001:0501:b1f9:0000:0000:0000:0000:0030
rdlength : 16
ttl : 172800
type : AAAA
}
{
class : 1
name : i.gtld-servers.net
rdata : 30.172.43.192
rdlength : 4
ttl : 172800
type : A
}
{
class : 1
name : i.gtld-servers.net
rdata : 2001:0503:39c1:0000:0000:0000:0000:0030
rdlength : 16
ttl : 172800
type : AAAA
}
{

```



```

class : 1
name : f.gtld-servers.net
rdata : 30.51.35.192
rdlength : 4
ttl : 172800
type : A
}
{
class : 1
name : a.gtld-servers.net
rdata : 30.6.5.192
rdlength : 4
ttl : 172800
type : A
}
=====
connecting to 30.6.5.192
=====
HEADER
=====
{
additional count : 8
answer count : 0
authority count : 5
id : 19258
is authoritative : False
is response : True
is truncated : False
opcode : 0
question count : 1
recursion available : False
recursion desired : False
reserved : 0
response code : No Error
}
=====
QUESTION
=====
{
Domain Name : yahoo.com
Query Class : 1
Query Type : 255
}
=====
ANSWER
=====
=====
AUTHORITY
=====
{
class : 1
name : yahoo.com
rdata : ns1.yahoo.com
rdlength : 6

```

```

ttl : 172800
type : NS
}
{
class : 1
name : yahoo.com
rdata : ns5.yahoo.com
rdlength : 6
ttl : 172800
type : NS
}
{
class : 1
name : yahoo.com
rdata : ns2.yahoo.com
rdlength : 6
ttl : 172800
type : NS
}
{
class : 1
name : yahoo.com
rdata : ns3.yahoo.com
rdlength : 6
ttl : 172800
type : NS
}
{
class : 1
name : yahoo.com
rdata : ns4.yahoo.com
rdlength : 6
ttl : 172800
type : NS
}
=====
ADDITIONAL
=====
{
class : 1
name : ns1.yahoo.com
rdata : 2001:4998:0130:0000:0000:0000:1001
rdlength : 16
ttl : 172800
type : AAAA
}
{
class : 1
name : ns1.yahoo.com
rdata : 16.131.180.68
rdlength : 4
ttl : 172800
type : A
}

```

```

{
  class : 1
  name : ns5.yahoo.com
  rdata : 124.247.160.119
  rdlength : 4
  ttl : 172800
  type : A
}
{
  class : 1
  name : ns2.yahoo.com
  rdata : 2001:4998:0140:0000:0000:0000:1002
  rdlength : 16
  ttl : 172800
  type : AAAA
}
{
  class : 1
  name : ns2.yahoo.com
  rdata : 16.255.142.68
  rdlength : 4
  ttl : 172800
  type : A
}
{
  class : 1
  name : ns3.yahoo.com
  rdata : 53.221.84.203
  rdlength : 4
  ttl : 172800
  type : A
}
{
  class : 1
  name : ns3.yahoo.com
  rdata : 2406:8600:00b8:fe03:0000:0000:0000:1003
  rdlength : 16
  ttl : 172800
  type : AAAA
}
{
  class : 1
  name : ns4.yahoo.com
  rdata : 157.11.138.98
  rdlength : 4
  ttl : 172800
  type : A
}
=====
connecting to 16.255.142.68
=====
HEADER
=====
{

```

additional count : 4
answer count : 16
authority count : 0
id : 6505
is authoritative : True
is response : True
is truncated : False
opcode : 0
question count : 1
recursion available : False
recursion desired : False
reserved : 0
response code : No Error
}

=====

QUESTION

=====

{
Domain Name : yahoo.com
Query Class : 1
Query Type : 255
}

=====

ANSWER

=====

{
class : 1
name : yahoo.com
rdata :
{
Admin MB : hostmaster.yahoo-inc.com
Expiration Limit : 1814400
Minimum TTL : 600
Primary NS : ns1.yahoo.com
Refresh interval : 3600
Retry interval : 300
Serial Number : 2017101223
}
rdlength : 49
ttl : 1800
type : SOA
}
{
class : 1
name : yahoo.com
rdata : 45.36.190.206
rdlength : 4
ttl : 1800
type : A
}
{
class : 1
name : yahoo.com
rdata : 109.253.138.98

```

rdlength : 4
ttl : 1800
type : A
}
{
class : 1
name : yahoo.com
rdata : 149.180.139.98
rdlength : 4
ttl : 1800
type : A
}
{
class : 1
name : yahoo.com
rdata : 2001:4998:0058:0c02:0000:0000:0000:00a9
rdlength : 16
ttl : 1800
type : AAAA
}
{
class : 1
name : yahoo.com
rdata : 2001:4998:0044:0204:0000:0000:0000:00a7
rdlength : 16
ttl : 1800
type : AAAA
}
{
class : 1
name : yahoo.com
rdata : 2001:4998:000c:0a06:0000:0000:0002:4008
rdlength : 16
ttl : 1800
type : AAAA
}
{
class : 1
name : yahoo.com
rdata :
{
Mail Exchanger : mta7.am0.yahoodns.net
Preference : 1
}
rdlength : 25
ttl : 1800
type : MX
}
{
class : 1
name : yahoo.com
rdata :
{
Mail Exchanger : mta5.am0.yahoodns.net

```

```

Preference : 1
}
rdlength : 9
ttl : 1800
type : MX
}
{
class : 1
name : yahoo.com
rdata :
{
Mail Exchanger : mta6.am0.yahoodns.net
Preference : 1
}
rdlength : 9
ttl : 1800
type : MX
}
{
class : 1
name : yahoo.com
rdata : ns3.yahoo.com
rdlength : 6
ttl : 172800
type : NS
}
{
class : 1
name : yahoo.com
rdata : ns5.yahoo.com
rdlength : 6
ttl : 172800
type : NS
}
{
class : 1
name : yahoo.com
rdata : ns2.yahoo.com
rdlength : 6
ttl : 172800
type : NS
}
{
class : 1
name : yahoo.com
rdata : ns1.yahoo.com
rdlength : 2
ttl : 172800
type : NS
}
{
class : 1
name : yahoo.com
rdata : ns4.yahoo.com

```

```
rdlength : 6
ttl : 172800
type : NS
}
{
class : 1
name : yahoo.com
rdata : #v=spf1 redirect=_spf.mail.yahoo.com
rdlength : 36
ttl : 1800
type : TXT
}
```

=====

AUTHORITY

=====

=====

ADDITIONAL

=====

```
{
class : 1
name : ns1.yahoo.com
rdata : 16.131.180.68
rdlength : 4
ttl : 1209600
type : A
}
```

```
{
class : 1
name : ns2.yahoo.com
rdata : 16.255.142.68
rdlength : 4
ttl : 1209600
type : A
}
```

```
{
class : 1
name : ns3.yahoo.com
rdata : 53.221.84.203
rdlength : 4
ttl : 1209600
type : A
}
```

```
{
class : 1
name : ns4.yahoo.com
rdata : 157.11.138.98
rdlength : 4
ttl : 1209600
type : A
}
```

=====

و جوابی که به درخواست‌کننده می‌فرستید باید به فرمت زیر باشد (مخصوصاً متن نمی‌فرستید و به فرمت پکت می‌فرستید!)

HEADER

```

=====
{
additional count : 0
answer count : 1
authority count : 0
id : 7
is authoritative : False
is response : True
is truncated : False
opcode : 0
question count : 0
recursion available : False
recursion desired : False
reserved : 0
response code : No Error
}

```

=====

QUESTION

=====

=====

ANSWER

=====

```

{
class : 1
name : yahoo.com
rdata : 45.36.190.206
rdlength : 4
ttl : 3600
type : A
}

```

=====

AUTHORITY

=====

=====

ADDITIONAL

=====

به عنوان خروجی فایل برای درخواست برعکس هم اگر آی دی ۱ و آی پی 98.138.11.157 بودند فایل به شکل زیر می شد.

connecting to 4.0.41.198

=====

HEADER

=====

```

{
additional count : 12
answer count : 0
authority count : 6
id : 42437
is authoritative : False
is response : True
is truncated : False
opcode : 0
question count : 1
recursion available : False

```



```

recursion desired : False
reserved : 0
response code : No Error
}
=====
QUESTION
=====
{
Domain Name : in.98.138.11.157-addr.arpa
Query Class : 1
Query Type : 255
}
=====
ANSWER
=====
AUTHORITY
=====
{
class : 1
name : in-addr.arpa
rdata : e.in-addr-servers.arpa
rdlength : 20
ttl : 172800
type : NS
}
{
class : 1
name : in-addr.arpa
rdata : f.in-addr-servers.arpa
rdlength : 4
ttl : 172800
type : NS
}
{
class : 1
name : in-addr.arpa
rdata : d.in-addr-servers.arpa
rdlength : 4
ttl : 172800
type : NS
}
{
class : 1
name : in-addr.arpa
rdata : c.in-addr-servers.arpa
rdlength : 4
ttl : 172800
type : NS
}
{
class : 1
name : in-addr.arpa
rdata : b.in-addr-servers.arpa

```

```

rdlength : 4
ttl : 172800
type : NS
}
{
class : 1
name : in-addr.arpa
rdata : a.in-addr-servers.arpa
rdlength : 4
ttl : 172800
type : NS
}
=====
ADDITIONAL
=====
{
class : 1
name : e.in-addr-servers.arpa
rdata : 101.86.119.203
rdlength : 4
ttl : 172800
type : A
}
{
class : 1
name : e.in-addr-servers.arpa
rdata : 2001:0dd8:0006:0000:0000:0000:0101
rdlength : 16
ttl : 172800
type : AAAA
}
{
class : 1
name : f.in-addr-servers.arpa
rdata : 1.9.0.193
rdlength : 4
ttl : 172800
type : A
}
{
class : 1
name : f.in-addr-servers.arpa
rdata : 2001:067c:00e0:0000:0000:0000:0000:0001
rdlength : 16
ttl : 172800
type : AAAA
}
{
class : 1
name : d.in-addr-servers.arpa
rdata : 53.60.10.200
rdlength : 4
ttl : 172800
type : A
}

```

```

}
{
  class : 1
  name : d.in-addr-servers.arpa
  rdata : 2001:13c7:7010:0000:0000:0000:0053
  rdlength : 16
  ttl : 172800
  type : AAAA
}
{
  class : 1
  name : c.in-addr-servers.arpa
  rdata : 10.169.216.196
  rdlength : 4
  ttl : 172800
  type : A
}
{
  class : 1
  name : c.in-addr-servers.arpa
  rdata : 2001:43f8:0110:0000:0000:0000:0010
  rdlength : 16
  ttl : 172800
  type : AAAA
}
{
  class : 1
  name : b.in-addr-servers.arpa
  rdata : 183.183.253.199
  rdlength : 4
  ttl : 172800
  type : A
}
{
  class : 1
  name : b.in-addr-servers.arpa
  rdata : 2001:0500:0087:0000:0000:0000:0087
  rdlength : 16
  ttl : 172800
  type : AAAA
}
{
  class : 1
  name : a.in-addr-servers.arpa
  rdata : 73.0.212.199
  rdlength : 4
  ttl : 172800
  type : A
}
{
  class : 1
  name : a.in-addr-servers.arpa
  rdata : 2001:0500:0013:0000:0000:0000:0073
  rdlength : 16

```

```

ttl : 172800
type : AAAA
}
=====
connecting to 1.9.0.193
=====
HEADER
=====
{
additional count : 0
answer count : 0
authority count : 6
id : 45427
is authoritative : False
is response : True
is truncated : False
opcode : 0
question count : 1
recursion available : False
recursion desired : False
reserved : 0
response code : No Error
}
=====
QUESTION
=====
{
Domain Name : in.98.138.11.157-addr.arpa
Query Class : 1
Query Type : 255
}
=====
ANSWER
=====
=====
AUTHORITY
=====
{
class : 1
name : in.98-addr.arpa
rdata : r.arin.net
rdlength : 12
ttl : 86400
type : NS
}
{
class : 1
name : in.98-addr.arpa
rdata : u.arin.net
rdlength : 4
ttl : 86400
type : NS
}
{

```

```

class : 1
name : in.98-addr.arpa
rdata : x.arin.net
rdlength : 4
ttl : 86400
type : NS
}
{
class : 1
name : in.98-addr.arpa
rdata : y.arin.net
rdlength : 4
ttl : 86400
type : NS
}
{
class : 1
name : in.98-addr.arpa
rdata : z.arin.net
rdlength : 4
ttl : 86400
type : NS
}
{
class : 1
name : in.98-addr.arpa
rdata : arin.authdns.ripe.net
rdlength : 20
ttl : 86400
type : NS
}
=====
ADDITIONAL
=====
connecting to arin.authdns.ripe.net
=====
HEADER
=====
{
additional count : 0
answer count : 0
authority count : 5
id : 24232
is authoritative : False
is response : True
is truncated : False
opcode : 0
question count : 1
recursion available : False
recursion desired : False
reserved : 0
response code : No Error
}

```

=====

QUESTION

=====

```
{
Domain Name : in.98.138.11.157-addr.arpa
Query Class : 1
Query Type : 255
}
```

=====

ANSWER

=====

=====

AUTHORITY

=====

```
{
class : 1
name : in.98.138-addr.arpa
rdata : ns1.yahoo.com
rdlength : 15
ttl : 86400
type : NS
}
{
class : 1
name : in.98.138-addr.arpa
rdata : ns2.yahoo.com
rdlength : 6
ttl : 86400
type : NS
}
{
class : 1
name : in.98.138-addr.arpa
rdata : ns3.yahoo.com
rdlength : 6
ttl : 86400
type : NS
}
{
class : 1
name : in.98.138-addr.arpa
rdata : ns4.yahoo.com
rdlength : 6
ttl : 86400
type : NS
}
{
class : 1
name : in.98.138-addr.arpa
rdata : ns5.yahoo.com
rdlength : 6
ttl : 86400
type : NS
}
}
```

```

=====
ADDITIONAL
=====
connecting to ns1.yahoo.com
=====
HEADER
=====
{
additional count : 8
answer count : 1
authority count : 5
id : 26939
is authoritative : True
is response : True
is truncated : False
opcode : 0
question count : 1
recursion available : False
recursion desired : False
reserved : 0
response code : No Error
}
=====
QUESTION
=====
{
Domain Name : in.98.138.11.157-addr.arpa
Query Class : 1
Query Type : 255
}
=====
ANSWER
=====
{
class : 1
name : in.98.138.11.157-addr.arpa
rdata : ns4.yahoo.com
rdlength : 15
ttl : 1800
type : PTR
}
=====
AUTHORITY
=====
{
class : 1
name : in.98.138.11-addr.arpa
rdata : ns1.yahoo.com
rdlength : 6
ttl : 172800
type : NS
}
{

```

```

class : 1
name : in.98.138.11-addr.arpa
rdata : ns5.yahoo.com
rdlength : 6
ttl : 172800
type : NS
}
{
class : 1
name : in.98.138.11-addr.arpa
rdata : ns2.yahoo.com
rdlength : 6
ttl : 172800
type : NS
}
{
class : 1
name : in.98.138.11-addr.arpa
rdata : ns4.yahoo.com
rdlength : 2
ttl : 172800
type : NS
}
{
class : 1
name : in.98.138.11-addr.arpa
rdata : ns3.yahoo.com
rdlength : 6
ttl : 172800
type : NS
}
=====
ADDITIONAL
=====
{
class : 1
name : ns1.yahoo.com
rdata : 16.131.180.68
rdlength : 4
ttl : 1209600
type : A
}
{
class : 1
name : ns2.yahoo.com
rdata : 16.255.142.68
rdlength : 4
ttl : 1209600
type : A
}
{
class : 1
name : ns3.yahoo.com
rdata : 53.221.84.203

```



```

rdlength : 4
ttl : 1209600
type : A
}
{
class : 1
name : ns4.yahoo.com
rdata : 157.11.138.98
rdlength : 4
ttl : 1209600
type : A
}
{
class : 1
name : ns5.yahoo.com
rdata : 124.247.160.119
rdlength : 4
ttl : 1209600
type : A
}
{
class : 1
name : ns1.yahoo.com
rdata : 2001:4998:0130:0000:0000:0000:1001
rdlength : 16
ttl : 86400
type : AAAA
}
{
class : 1
name : ns2.yahoo.com
rdata : 2001:4998:0140:0000:0000:0000:1002
rdlength : 16
ttl : 86400
type : AAAA
}
{
class : 1
name : ns3.yahoo.com
rdata : 2406:8600:00b8:fe03:0000:0000:1003
rdlength : 16
ttl : 86400
type : AAAA
}
=====

```

و پکت پاسخ هم به شکل زیر است.

```

HEADER
=====
{
additional count : 0
answer count : 1
authority count : 0
id : 1

```

```

is authoritative : False
is response : True
is truncated : False
opcode : 0
question count : 0
recursion available : False
recursion desired : False
reserved : 0
response code : No Error
}
=====
QUESTION
=====
ANSWER
=====
{
class : 1
name : in.98.138.11.157-addr.arpa
rdata : ns4.yahoo.com
rdlength : 15
ttl : 3600
type : PTR
}
=====
AUTHORITY
=====
ADDITIONAL
=====
=====

```

۳.۰ نکات دیگر

- در صورتی که پکت ورودی مشکل داشت یا در حین انجام کار به مشکلی برخوردید (مثلاً هیچ آی‌پی پیدا نشد یا نام آی‌پی‌ها با سوال یکی نبود) باید یک جواب از نوع refused بدهید

```

HEADER
=====
{
additional count : 0
answer count : 0
authority count : 0
id : 4
is authoritative : False
is response : True
is truncated : False
opcode : 0
question count : 0
recursion available : False
recursion desired : False
reserved : 0
response code : Refused
}

```

```

}
=====
QUESTION
=====
ANSWER
=====
AUTHORITY
=====
ADDITIONAL
=====
=====

```

- در کنار برنامه‌ی خود یک فایل اسکریپتی به اسم dns.sh قرار دهید که یک آرگومان ورودی بگیرد که همان آی‌پی سرور ریشه‌است در این فایل کدهای لازم برای کامپایل و اجرا شدن کد خود را قرار دهید ، در حین جاج فقط از این فایل استفاده خواهد شد.
- از لینک بالا فرمت متنی مخصوص قسمت name را که در بعضی جاهای دیگر هم استفاده شده، مطالعه کنید.
- به فرمت خروجی و پکت‌ها دقت کنید.
- توجه کنید که برنامه‌ی شما، چیزی در خروجی استاندارد نمی‌نویسد و فقط پکت پاسخ و فایل‌ها بررسی می‌شوند.
- فایل‌های خود را در یک پوشه بگذارید و فایل dns.sh حتما در ریشه باشد (منظور از ریشه همان فولدر اصلی است)
- فایل‌های لاگ باید در همان ریشه‌ی پروژه قرار بگیرند.
- برای درخواست‌هایی که qr یا opcode ایراد دارند و جواب آن‌ها بدون هیچ کار دیگری refused است فایل لاگ تولید نکنید.
- برنامه‌ی خود را به شکل چندریشه‌ای^۷ پیاده‌سازی کنید. دقت داشته باشید که اگر به هر دلیلی برنامه‌ی شما با یکی از درخواست‌ها کار نکرد و به مشکل خورد، متوقف نشود. زیرا در حین جاج برنامه‌ی شما یک بار اجرا خواهد شد و اگر در یکی از درخواست‌ها متوقف شود و به درخواست‌های بعدی پاسخ ندهد، نمره‌ی درخواست‌های بعدی را هم نخواهید گرفت.
- برای تحویل در گیت درس در همان فولدر تمرین قبلی با نام شماره دانشجویی خود یک فولدر دیگر به اسم hw1 بسازید و کدهای خود را در آن دایرکتوری قرار دهید به طوری که dns.sh در ریشه همان فولدر باشد.
- می‌توانید از زبان‌های پایتون ، C ، ++C و جاوا استفاده کنید. فقط مطمئن شوید بعد از اجرای dns.sh کد شما کامپایل و بعد اجرا می‌شود. ضمناً فقط از کتابخانه‌ها و منابع دیگر استفاده نکنید.
- در این تمرین و سایر تمرینهای درس، با هرگونه تقلب شدیداً برخورد خواهد شد.
- کدها باید توسط خود شما نوشته شوند و کپی کردن از منابع اینترنتی تقلب محسوب می‌شود.
- در صورت داشتن هرگونه سوال از طریق piazza اقدام کنید.
- برای تحویل این تمرین تا ساعت ۲۳:۵۹ روز ۸ آبان فرصت دارید.

موفق باشید

multi-thread^۷