ЛАБОРАТОРНА РОБОТА № 8

Тема: Вивчення системи DNS. Утиліта NSLOOKUP
Мета: Вивчити принципи та призначення системи DNS

Теоретична частина

Комп'ютери, що мають підключення до комп'ютерної мережі часто називають хостами (від англ. host), а для їх ідентифікації у мережі використовують *імена хоста.* Ім'я хоста – це певний мнемонічний, зручний для сприйняття людиною запис. Наприклад, znu.edu.ua, cnn.com, www.yahoo.com, gaia.cs.umass.edu, surf.eurecom.fr. У той же час для комп'ютерів прийнятними є тільки числові форми ідентифікації. Зараз для ідентифікації хостів в Інтернет широко використовується IP-адреса протоколу IPv4 – сукупність чотирьох однобайтових чисел з певною ієрархічною структурою. Символьне ім'я хоста є зручним для людини, вони легко запам'ятовується, але алгоритми обробки IP-адреси є більш простими для комп'ютерних систем, мереж та маршрутизаторів. Також важливим є те, що зв'язок (асоціація) між IP-адресою та символьним ім'ям робиться не назавжди. Зміна провайдера Інтернет або хостингової компанії буде змінювати IP-адресу але символьне (доменне) ім'я ресурсу, наприклад, вебсайту якоїсь організації скоріш за все буде залишатись тим самим, якщо, безумовно, ця організація не захоче його змінити. Крім того сам провайдер може інколи змінювати доступні йому діапазони IP-адрес, що також викликає зміну IP-адрес у споживачів його сервесів. Тому принципово важливим є здатність системи до зберігання символьної адреси при ймовірній зміни IP-адреси.

На початку створення та існування Internet (ARPANET) її зростання було помірним, тому реляція символьних імен хостів та IP-адреси могла централізовано контролюватись та підтримуватись Мережевим Інформаційним Центром (NIC – Network Information Center) за допомогою єдиного файлу (hosts.txt). Адміністратор будь-якого хоста або певної організації періодично робив копію цього файлу з використанням або електронного передавання файлів, або навіть якогось носія, наприклад, дискети. Нові ресурси додавались не часто, тому цей файл був невеликим і міг не змінюватись навіть декілька тижнів. Механізм визначення мережевої адреси за символьним ім'ям хоста полягав у звернені до файлу hosts.txt, свіжа копія якого переносилась у певний каталог на кожний комп'ютер, що підключався до мережі. Така система працювала достатньо добре поки ARPANET була порівняно невеликою мережею. Коли темп зростання та зміни складу мережі суттєво збільшились, частота звернення за оновленим файлом hosts.txt із NIC також суттєво зросла. Крім того стало необхідним відокремити управління локальними іменами і адресами у різних організаціях та компаніях, а також постійно вносити зміни до файлу hosts.txt. Тому централізована схема підтримки файлу із записами відповідності імен хостів та IP-адрес стала недостатньо практичною та повільною і потребувала

зміни принципів у цьому питані. У 1983 році Пол Мокапетріс — науковець із Каліфорнійського університету, запропонував створити автоматизовану систему доменних імен DNS (Domain Name System).

Перші результати розробки DNS було опубліковано у 1983 році в RFC 882 та 883. Після експериментів з декількома реалізаціями, DNS було формально визначено у RFC 1034 та 1035 у 1987 році. Положення, специфікації, протоколи та принципи використання DNS продовжують розвиватись і у наш час, що закріплюється відповідними стандартами у документах RFC (https://datatracker.ietf.org/).

DNS базується на двох основних концепціях:

- на розподіленій базі даних, що зберігає узагальнені записи про ресурси мережі (resource records) та має децентралізоване управління;
- на певній схемі іменування, зо базується ієрархічно структурованих доменних іменах.

DNS є розподіленою базою даних. Це дозволяє локально контролювати окремі сегменти загальної бази даних. Дані у кожному сегменті мають доступ через мережу з використанням технології клієнт-сервер. Адекватна продуктивність досягається через використання механізмів копіювання (replication) та кешування (caching).

Програми, що реалізують серверну частину DNS, звуться серверами імен (name servers). Сервер імен має інформацію про деякий сегмент загальної бази даних DNS, для якого він є повноважним сервером, та робить її доступною для клієнтів — програми вирішувачі (resolvers). Програми вирішувачі (resolvers) зазвичай надають бібліотечні функції, які генерують запити та насилають їх через мережу до серверів імен. DNS-перетворювач виконує дві основні функції. Функція *gethostbyname()* перетворює доменне ім’я у IP-адресу, а функція *gethostbyaddr()* перетворює IP-адресу у доменне ім’я. Перетворювач взаємодіє з одним або декількома DNS-серверами для виконання цих функцій від імені додатку. Для виконання цього процесу перетворювач повинен мати налаштування принаймні на один DNS-сервер. Для комп’ютера в мережі Internet зазвичай вказується список IP-адрес, кожна з яких відповідає певному DNS-серверу. Мережеві адміністратори намагаються розташовувати локальні DNS-сервери ближче до клієнтів, що використовують їх для запитів. Наприклад, в університеті комп’ютерна мережа, що об’єднує комп’ютери класів, кафедр, деканатів, відділів, також має у своєму складі локальні DNS-сервери, що оброблюють запити вирішувачів від цих комп’ютерів. Зазвичай провайдер Інтернет також розташовує свої DNS-сервери у мережі, до якої підключаються його клієнти.

Кожна одиниця даних розподіленої бази даних DNS індексується за ім’ям. Ці імена, за своєю суттю, є своєрідними вказівниками в інвертованому дереві простору доменних імен.

Кожний вузол (node) у просторі імен має власну мітку без крапки тому, що крапка (".") використовується в якості роздільників міток. Максимальна довжина мітки може складати 63 байта. Максимальна довжина доменного

імені (сума усіх міток та роздільників) дорівнює 255 байтам. Кореневий домен має мітку нульової довжини (null). Повне доменне ім’я кожного вузлу у дереві — це послідовність міток на шляху від цього вузла до кореня (Рис.1). За згодою, мітки, що складають доменне ім’я читають зліва на право, починаючи з нижньої, найбільш віддаленої від кореня, і завершуючи самою верхньою, яка безпосередньо знаходиться перед коренем.

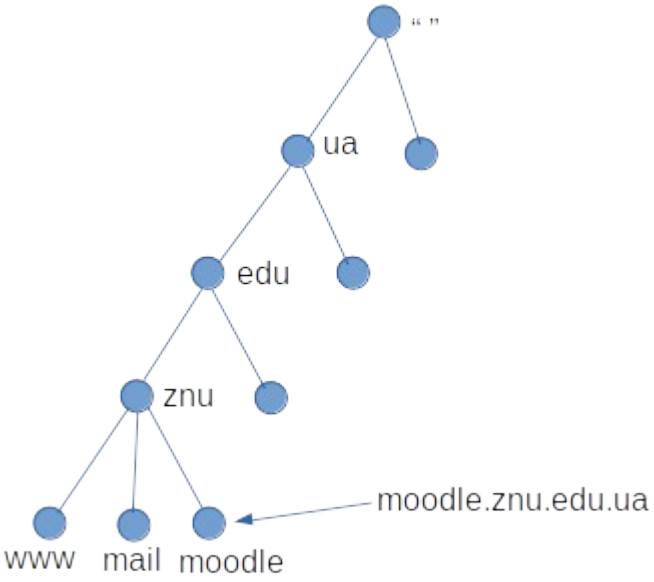


Рис.1. Приклад побудови доменного імені.

Для позначення кореневого домену в доменному імені вузла використовується символ крапки (".") наприкінці імені. Доменні імена, які закінчуються крапкою, звуться абсолютними доменними іменами (Рис.2).

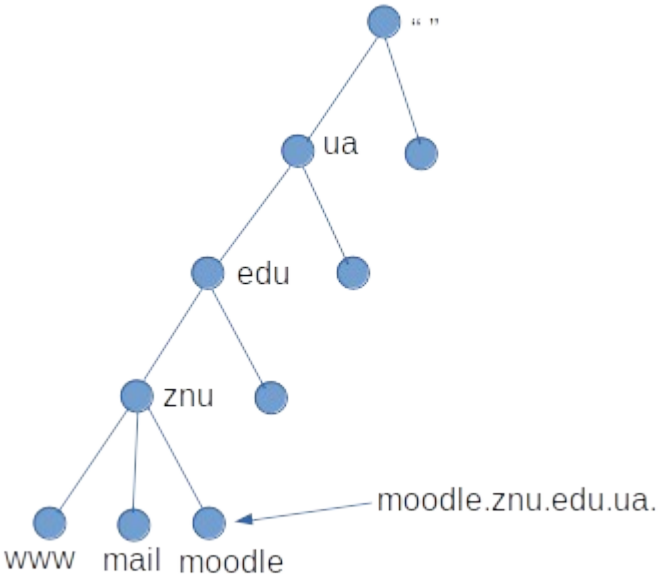


Рис.2. Приклад абсолютного доменного імені.

Абсолютне доменне ім’я зв’язане з коренем, тому воно однозначно специфікує місце знаходження вузла в ієрархії. Імена без крапки наприкінці

інколи інтерпретуються як ті, що пов’язані з деяким доменом, який відрізняється від кореневого. Абсолютне доменне ім’я також зветься повністю кваліфікованим доменним ім’ям (fully-qualifed domain name або FQDN).

Домен - це деяке піддерево простору доменних імен. Доменне ім’я домену – це ім’я самого верхнього вузлу домену. Наприклад, верхнім вузлом домену "znu.edu.ua" є вузол, який має доменне ім’я "znu.edu.ua" (Рис.3).

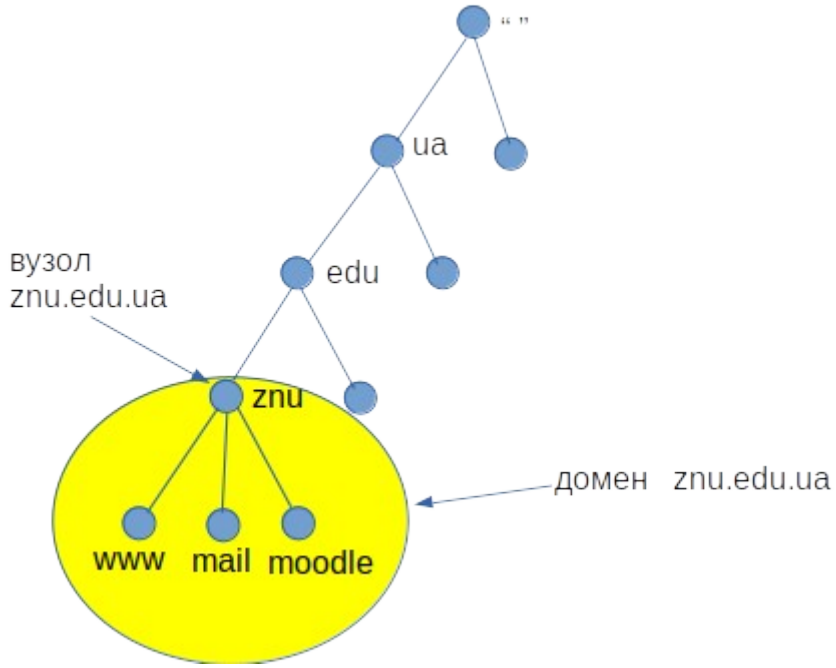


Рис.3. Верхній вузол в ієрархії має ім’я домену.

Кожне доменне ім’я може знаходитись у декількох піддеревах і, відповідно, у декількох доменах. Наприклад, доменне ім’я "moodle.znu.edu.ua" є частиною доменів "znu.edu.ua", "edu.ua", а також домену "ua". Хост-машини також є вузлами DNS-дерева, і також, за суттю доменної ієрархії, є доменами. Домен має усі хост-машини, доменні імена яких знаходяться всередині цього домену, тобто домен є суфіксом у їх іменах. Хост-машини у домені зв’язані логічно (інколи географічно або за організаційною ознакою) і не зобов’язані належати до певної мережі або класу IP-адрес. В одному домені можуть знаходиться хост-машини із різних мереж, різних стран і навіть різних континентів. Доменні імена всередині домену можуть бути іменами хост-машин або посилатись на структурну інформацію про піддомени. Одне і те саме доменне ім’я може використовуватись як для позначення хост-машини, так і для домену. Наприклад, ім’я "hp.com" є іменем домену компанії Hewlett-Packard і іменем хост-машини, яка здійснює маршрутизацію пошти між HP та Internet. Або, ім’я "znu.edu.ua" є іменем домену компанії та іменем хост-машини, яка отримує запити на веб-сервер університету. Тип інформації, яку необхідно знайти про хост-машину або піддомен, залежить від контексту використання доменного імені.

Простір імен DNS має ієрархічну структуру і містить множину вкладених доменів, кожен з яких представляє адміністративно пов'язаний набір хост-машин Internet. Безпосередньо нижче кореня цієї ієрархії знаходиться набір доменів верхнього рівня, які відповідають або певним типам організацій, або пов’язані з географічним розташуванням ресурсу. Географічне розташування ресурсу є не стільки фізичним, скільки логічним з точки зору формуванням доменного імені. Наприклад, такий домен верхнього рівня як “tv” є географічним, позначає державу Тувалу і належить її уряду. Але цей домен за контрактами з урядом Тувалу використовується в усьому світі телевізійними та медійними компаніями, реальні ресурси яких розташовуються не в Тувалу.

Дані, які асоціюються з доменними іменами, знаходяться у ресурсних записах (resouce records або RR). Записи поділяються на класи. У сучасній DNS практично використовується єдиний клас – IN (Internet). Також записи поділяються на типи. Тип записи визначає тип інформації, яка може зберігатись у просторі доменних імен. Кожний тип записи має визначений формат. В Internet-класі є адресні записи, записи серверів імен, записи вказівників та інші. Серед них є дві обов’язкові, відповідні записи повинні бути у будь-якій зоні, інакше вона буде важитись некоректною. Обов’язковими типами є SOA та NS.

Ресурсний запис є одиницею зберігання та передавання інформації в DNS і має наступні поля:

- ім’я (Name), ім’я домену, до якого відноситься цей запис;
- TTL (Time Ti Live), заданий час зберігання запису неавторизованим сервером;
- тип (Type), параметр, який визначає призначення та формат запису в полі даних (Rdata);
- клас (Class), тип мережі передавання даних;
- довжина поля даних (Rdlen);
- поле даних (Rdata), вміст та формат залежать від типу запису.

Найчастіше використовуються такі типи ресурсних записів:

- SOA (Start of Authority), "початок повноважень" або початок зони відповідальності, вміщує базові параметри доменної зони: ім’я первинного DNS-серверу та контактну адресу електронної пошти;
- A (IPv4 Address Record), адресний запис, який зв’язує доменне ім’я з IPv4 адресою хоста;
- AAAA (IPv6 Address Record), адресний запис, який зв’язує доменне ім’я з IPv6 адресою хоста;
- CNAME (Canonical Name Record), канонічний запис імені, який використовується для перенаправлення на інше доменне ім’я;
- MX (Mail Exchange) поштовий обмінник, який посилається на поштовий сервер, що обслуговує домен;
- NS (Name Server), сервер імен, що посилається на DNS-сервер, який відповідає за домен;

- TXT, текстове описання домену, часто необхідно для виконання специфічних задач, наприклад, підтвердження права власності на домен при прив’язуванні його до поштового сервісу;
- PTR (Point to Reverse), запис вказівки, що зв’язує IP адресу хоста з доменом, часто використовується поштовими сервісами.

З 1995 р. в основі DNS знаходиться система кореневих (root) серверів, яка складається з первинного (primary) сервера a.root-servers.net та його реплік (secondary). З 1997 і по сьогодення таких реплік 12, з іменами b.root-servers.net, c.root-servers.net і т.д. до m.root-servers.net. Їх список, розподіл у світі та інша інформація доступна на офіційному сайті www.root-servers.org. Коли було створено «скритий» мастер-сервер a.root-servers.net став одним із 13 рівноправних кореневих серверів, які управляються 12 незалежними організаціями-операторами, що несуть за них відповідальність:

a — VeriSign Global Registry Services;
b — University of Southern California, Information Sciences Institute;
c — Cogent Communications;
d — University of Maryland;
e — NASA Ames Research Center;
f — Internet Systems Consortium, Inc.;
g — US DoD Network Information Center;
h — US Army Research Lab;
i — Netnod;
j — VeriSign Global Registry Services;
k — RIPE NCC;
l — ICANN;
m — WIDE Project.

З метою підвищення продуктивності та стійкості системи, у відповідь на вибухове зростання інтернету і кількості звернень до доменів, з 2003 року було впроваджено технологію anycast, що дозволила операторам зробити множину дзеркал кореневих серверів, які розташовуються ближче до користувачів. Зараз число таких дзеркал становить 1467 (https://root-servers.org/).

Кореневі сервери DNS є критичним компонентом системи, тому що забезпечують доступ до кореневої зони DNS. Корнева зона має інформацію про усі домени верхнього (першого) рівня. Ця інформація вказує клієнту на які сервери DNS слід відправити запит для визначення повного доменного імені. Якщо інформація про домен у запиті не була раніше збережена у кеші клієнта, то його запит починається із звернення до кореневого серверу і буде обробленим найближчим його дзеркалом.

Якщо є необхідність змінити адресацію у кореневій зоні, наприклад, змінити склад серверів, які обслуговують домен, то адміністратор домену верхнього рівня надсилає підписаний ЕЦП запит, який ретельно перевіряється та авторизується спеціальним аудитором - IANA. Авторизований запит передається оператору VeriSign, який має право

вносити зміни на скритому мастер-сервері DNS, після чого зміни розповсюджуються через захищений протокол на всі кореневі сервера.

Практична частина

Стандартною утилітою, що широко використовується для перевірки сервісів DNS-серверів, є **nslookup**. Утиліта **nslookup** інсталюється разом із протоколом TCP/IP і використовується із командного рядка. Використання nslookup.exe має декілька особливостей. Насамперед, у параметрах протоколу TCP/IP має бути вказаним принаймні один сервер DNS. Для визначення встановлених налаштувань DNS можна використати команду ipconfig /all (в операційній системі Windows), або команду resolvectl dns (в операційній системі Linux).

Утиліта nslookup може використовуватись у двох режимах: інтерактивному або неінтерактивному. Неінтерактивний режим використовується, якщо відповідь може бути у вигляді одного набору даних. Для запуску nslookup у неінтерактивному режимі використовується наступним синтаксис:
nslookup [-параметри] [вузол] [сервер]

Для запуску nslookup в інтерактивному режимі у командному рядку виконується команда nslookup:

в операційній системі Windows

```
C:\> nslookup
Default Server:  nameserver1.domain.com
Address:  10.0.0.1
>
```

в операційній системі Linux

```
(base) :~$ nslookup
> server
Default server: 127.0.0.53
Address: 127.0.0.53#53
>
```

Список доступних команд nslookup можна отримати за допомогою команди help або «?», а також в операційній системі Linux можна виконати man nslookup. Нижче наведено приклад результату виконання команди help:

```
Commands:   (ідентифікатори записано у верхньому регістрі, [] позначають
опції)

 NAME            - виводить інформацію про хост/домен NAME з використанням
                   серверу за замовчуванням
 NAME1 NAME2     - та сама операція, але використовується NAME2 як сервер
 help or ?       - виводить інформацію про стандартні команди
 set OPTION      - задає опції
```

```
    all                 - виводить параметри поточного серверу та хоста
    [no]debug           - виведення інформації налагодження
    [no]d2              - виведення повної інформації налагодження
    [no]defname         - додати ім’я домену до усіх запитів
    [no]recurse         - запит про рекурсивну відповідь на запит
    [no]search          - використовувати список пошуку доменів
    [no]vc              - завжди використовувати віртуальну схему
    domain=NAME         - встановити ім’я домену за замовчуванням NAME
    srchlist=N1[/N2/.../N6] — встановити домен N1 і список пошуку N1, N2, і
т.д.
    root=NAME           - встановити кореневий сервер NAME
    retry=X             - встановити число повторень X
    timeout=X           - встановити інтервал часу очікування X секунд
    type=X              - встановити тип запиту (наприклад, A, ANY, CNAME,
MX,
                          NS, PTR, SOA, SRV)
    querytype=X         - те саме, що і type
    class=X             - встановити клас запиту (наприклад, IN (Internet),
ANY)
    [no]msxfr           - використовувати швидку зону MS для передавання
    ixfrver=X           - поточна версія, що використовується в запитах IXFR

 server NAME     - встановити сервер за замовчуванням NAME, використовувати
                   поточний сервер за замовчування
 lserver NAME    - встановити сервер за замовчуванням NAME, використовувати
                   первинний сервер
 root            - зробити поточний сервер за замовчуванням кореневим
сервером
 ls [opt] DOMAIN [> FILE] — список адрес у домені DOMAIN (опціонально:
виведення
                   у FILE)

    -a          -  список канонічних імен та псевдонімів
    -d          -  список усіх записів
    -t TYPE     -  список записів вказаного типу (наприклад, A, CNAME,
                   MX, NS, PTR, and so on)

 view FILE       - сортування файлу 'ls' і вивід його вмісту за допомогою pg
 exit            - вихід із програми
```

Реалізація nslookup суттєво залежить від типу та версії операційної системи. Якщо дані, які введено у командному рядку, не відповідають синтаксису команди nslookup, то вони інтерпретуються як ім’я вузла і робиться спроба визначити це ім’я за допомогою серверу за замовчуванням. Для переривання виконання команди в інтерактивному режимі використовується комбінація CTRL+C, а для завершення роботи nslookup в інтерактивному режимі – використовується команда exit.

Для зміни параметрів використання nslookup застосовується команда set. Для визначення поточного значення параметрів використовується команда set all. Наприклад:

```
> set all
Default server: 127.0.0.53
Address: 127.0.0.53#53

Set options:
  novc                  nodebug         nod2
  search                recurse
  timeout = 0           retry = 3       port = 53       ndots = 1
  querytype = A         class = IN
  srchlist =
```

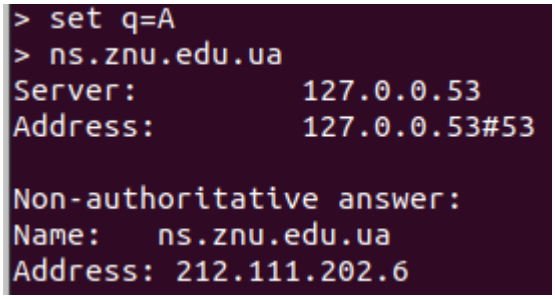
Для того, щоб знайти в адресному просторі домену дані різних типів, використовуються команди set type або set q[uerytype]. Наприклад:

```
> set q=mx
> znu.edu.ua
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
znu.edu.ua      mail exchanger = 10 mx1.znu.edu.ua.
znu.edu.ua      mail exchanger = 20 mx2.znu.edu.ua.
```

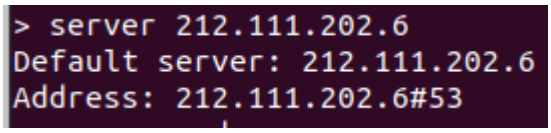
Позначення “Non-authoritative answer” вказує на те, що відповідь було отримано не від DNS-серверу зони відповідальності. Тобто така відповідь надходить від найближчого проміжного серверу, який отримав відповідний запит та мав у своєму кеші необхідний запис. Це відбувається тому, що записи про ресурси на DNS-серверах зони відповідальності залишаються незмінними тривалий час і кешування записів на інших DNS-серверах є звичайною практикою розвантаження DNS. Але, якщо потрібно отримати відповідь від DNS-серверу зони відповідальності, то необхідно задати ім’я потрібного сервера за допомогою команди server або lserver. Команда lserver визначає адрес сервера, на який потрібно відправляти запити, використовуючи локальний сервер. Команда server, за замовчуванням, використовує для отримання цієї адреси поточний сервер. Наприклад, якщо потрібно отримати автоматизовану відповідь для доменного імені www.znu.edu.ua, то першим визначимо DNS-сервер, що відповідає за зону znu.edu.ua:

```
> set q=NS
> znu.edu.ua
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
znu.edu.ua      nameserver = ns2.trifle.net.
znu.edu.ua      nameserver = ns.znu.edu.ua.
```

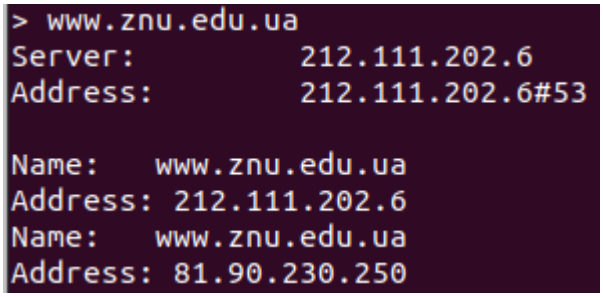
Наступним визначимо його IP-адресу:



Тепер за допомогою команди server змінюємо DNS-сервер, до якого будуть надсилатись запити:



Після цього усі запити будуть надсилатись саме до заданого DNS-серверу. Отримаємо відповідь для www.znu.edu.ua:



У відповіді відсутнє позначення “Non-authoritative answer”.

Утиліта nslookup також дозволяє отримати повний список вузлів домену за допомогою команди ls. Загальний синтаксис команд ls:

```
ls [- a | d | t type] domain [> filename]
```

Якщо команда ls виконується без аргументів, то буде отримано список усіх серверів - імені та адрес в домені. Параметр -a дозволяє отримати список канонічних імен і псевдонімів, параметр -d — отримати список усіх записів, а параметр -t – виконати фільтрацію за типом записів.

На деяких серверах DNS передавання зон дозволено тільки для авторизованих адрес або мереж. При спробі отримати данні зони з такого серверу з’явиться наступне повідомлення про помилку:

*** Can't list domain *example.com*.: Query refused

Завдання до лабораторної роботи

1. За допомогою утиліти nslookup визначте IP адреси 3 будь-яких Web-серверів в Інтернет. Занесіть у звіт інформацію у вигляді: DNS-ім’я - IP-адреса.
2. Командою set type=PTR (або set querytype=PTR) переведіть nslookup у режим трансляції IP адрес у DNS імена. Визначте імена серверів по наступним IP-адресам: 3.20.100.150; 128.84.21.199; 91.198.174.192; 212.8.40.1. Наведіть у звіті результат роботи програми.
3. Командою set type=NS (або set querytype=NS) переведіть nslookup у режим визначення DNS-імен по IP адресам та завантажте список кореневих серверів імен Інтернету, для чого в рядку команди задайте “.” (крапку). Результат занесіть у звіт.
4. Подібним чином отримайте список серверів імен домену ua, для чого задайте у якості команди “ua.” (ua точка). Результат занесіть у звіт.
5. Подібним чином отримайте список серверів імен домену zp.ua, для чого задайте у якості команди zp.ua. (zp.ua точка). Результат занесіть у звіт.
6. За допомогою nslookup виконайте визначення існуючого імені хосту в його IP адресу (наприклад, znu.edu.ua), одночасно захопіть та проаналізуйте пакети з повідомленнями DNS, для чого використовуйте програму WireSharck. Занесіть у звіт типи транспортних протоколів та номери портів, що використовуються у повідомленнях DNS. Занесіть до звіту зміст пакету запиту до DNS-серверу і його відповідь у вигляді: Ім’я поля - Розмір у байтах - Значення.
7. За допомогою nslookup виконайте визначення неіснуючого імені хосту в його IP адресу (наприклад, таку адресу: 123.znu.edu.ua), одночасно захоплюйте та аналізуйте пакети з повідомленнями DNS. Занесіть у звіт запит до DNS-серверу та його відповідь у вигляді: Ім’я поля - Розмір у байтах - Значення.
8. Виконайте трансляцію існуючої IP адреси у ім’я хосту (наприклад, 192.168.1.1), одночасно захоплюйте та аналізуйте пакети з повідомленнями DNS. Занесіть у звіт запит до DNS-серверу та його відповідь у вигляді: Ім’я поля - Розмір у байтах - Значення.
9. Виконайте трансляцію неіснуючої IP адреси у ім’я хосту (наприклад, 192.168.240.244), одночасно захоплюйте та аналізуйте пакети з повідомленнями DNS. Занесіть у звіт запит до DNS-серверу та його відповідь у вигляді: Ім’я поля - Розмір у байтах - Значення.

Контрольні запитання

1. Які RFC документи присвячено DNS системі? Які питання в них розглянуто?
2. Надайте пояснення наступних записів про ресурси: SOA, PTR, A, MX, CNAME.
3. Яке призначення утиліти nslookup?
4. Які порти використовуються для DNS сервера і клієнта?
5. Які транспортні протоколи використовуються у DNS системі? Якими RFC документами це регламентується?
6. Яке призначення і як працює програма вирішувач (resolver)?
7. Як іменуються root-сервера системи DNS?
8. Чим відрізняються авторитативна та неавторитативна відповіді?
9. Скільки рівнів DNS забезпечують ім’я www.znu.edu.ua? Надайте пояснення.
10. Наведіть приклад PTR запису та поясніть її.