

# **Instytut Teleinformatyki**

Wydział Inżynierii Elektrycznej i Komputerowej  
Politechnika Krakowska

**programowanie usług sieciowych**

---

***„Usługa DNS”***

laboratorium: 03

**Kraków, 2014**

## Spis treści

Spis treści .....	2
1. Wiadomości wstępne .....	3
1.1. Tematyka laboratorium .....	3
1.2. Zagadnienia do przygotowania .....	4
1.3. Cel laboratorium .....	4
2. Przebieg laboratorium .....	6
2.1. Przygotowanie laboratorium .....	6
2.2. Zadanie 1. Pierwszy program klienta DNS .....	6
2.3. Zadanie 2. Uzyskanie nazwy hosta z adresu IP. ....	7
2.4. Zadanie 3. Uzyskanie adresu IPv6 hosta. ....	7
2.5. Zadanie 4. Polecenie traceroute .....	7
2.6. Zadanie 5. Program wezdate .....	7
2.7. Zadanie 6. Typy rekordów bazy DNS .....	8
3. Opracowanie i sprawozdanie .....	9

# 1. Wiadomości wstępne

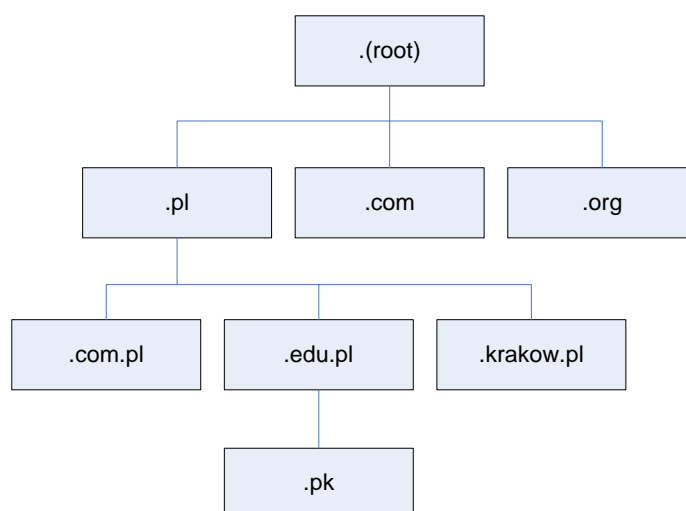
Pierwsza część niniejszej instrukcji zawiera podstawowe wiadomości teoretyczne dotyczące systemu nazw domenowych, czyli **DNS** (ang. *Domain Name System*). Poznanie tych wiadomości umożliwi prawidłowe zrealizowanie praktycznej części laboratorium.

## 1.1. Tematyka laboratorium

Tematyką laboratorium jest programowanie aplikacji wykorzystujących system **DNS**. Jest to system serwerów oraz protokół komunikacyjny zapewniający zamianę adresów znanych użytkownikom Internetu na adresy zrozumiałe dla sieci komputerowej. Dzięki wykorzystaniu **DNS** nazwa mnemoniczna, np. *www.pk.edu.pl* może zostać zamieniona na odpowiadający jej adres IP, czyli *149.156.132.156*.

Podstawą technicznego systemu **DNS** jest ogólnosiwiatowa sieć serwerów. Przechowują one informację na temat adresów domen. Każdy wpis zawiera nazwę oraz odpowiadający jej adres IP. Wpisy udostępniane są automatycznie, co pozwala na pracę Internetu.

DNS to również protokół komunikacyjny. Opisuje on sposób łączenia się klientów z serwerami DNS. Częścią specyfikacji protokołu jest również zestaw zaleceń, jak aktualizować wpisy w bazach domen internetowych.



**Rys. 1.** Przykład hierarchii DNS

## 1.2. Zagadnienia do przygotowania

Przed przystąpieniem do realizacji laboratorium należy zapoznać się z zagadnieniami dotyczącymi serwerów DNS:

- Struktura i działanie systemu DNS; [ Krysiak, Rozdział 7.1 ]
- Typy rekordów (ang. *resource records RR*) zasobów DNS: [ RFC 1034 ]
  - A – tłumaczy nazwę hosta na jego adres IPv4;
  - AAAA – jak A tylko dla adresów IPv6;
  - NS – wskazuje serwer dla danej strefy;
  - MX – wskazuje domyślny serwer poczty dla danej domeny;
  - CNAME – opisuje alias dla wcześniej zdefiniowanej nazwy hosta;
  - PTR – tłumaczy adres IP hosta na jego nazwę;
  - HINFO – zawiera opis sprzętu i systemu operacyjnego serwera;
  - SOA – potwierdza, że serwer jest wiarygodny dla danej strefy;
- Funkcje biblioteki DNS: [ Stevens ]
  - `gethostbyname()` - pobieranie adresu IP z nazwy hosta;
  - `gethostbyaddr()` - pobieranie nazwy hosta z adresu IP;
  - `gethostname()` - pobieranie nazwy hosta;
  - `uname()` - pobieranie informacji o systemie;
  - `gethostbyname2()` - jak `gethostbyname()` ale obsługuje IPv6;
  - `gethostent()` - czyta następną linię z `/etc/hosts`.

### Literatura:

- [1] W.R. Stevens, „*Programowanie Usług Sieciowych*”, „*API: gniazda i XTI*”.
- [2] K. Krysiak, „*Sieci komputerowe – kompendium*”.
- [3] „*DNS related RFCs*” (<http://www.dns.net/dnsrd/rfc/>).
- [4] RFC 1035, „*Domain Names--Implementation and Specification*”.
- [5] RFC 1034, „*Domain Names--Concepts and Facilities*”.

## 1.3. Cel laboratorium

Celem laboratorium jest poznanie działania systemu DNS i nauczenie pisania prostych programów wykorzystujących ten system. Podczas realizacji tego laboratorium zapoznasz się z:

- Architekturą rozproszonej bazy danych DNS;
- Mechanizmem działania klienta DNS;
- Funkcjami klienta DNS;
- Oraz opanujesz umiejętności programowania przy użyciu funkcji oraz struktur klienta DNS.



## 2. Przebieg laboratorium

Druga część instrukcji zawiera zadania do praktycznej realizacji, które demonstrują zastosowanie technik z omawianego zagadnienia.

### 2.1. Przygotowanie laboratorium

Przed przystąpieniem do dalszej części laboratorium należy skopiować całą zawartość katalogu `home/shared/pus/pus_03_dns` np. do katalogu `~/pus/pus_03_dns`.

### 2.2. Zadanie 1. Pierwszy program klienta DNS

Zadanie ma na celu pomóc zrozumieć zasadę działania funkcji `gethostbyname()` oraz zaprezentować strukturę `hostent`.

1. Najpierw należy przeanalizować i skompilować program `cw1_dns.c`:  

```
$ cc cw1_dns.c -o cw1_dns
```
2. Następnie uruchamiamy skompilowany program  

```
$ ./cw1_dns <nazwa_hosta>
```

np: 

```
$ ./cw1_dns www.o2.pl
```
3. Wynikiem działania tego programu jest wypisanie na ekran oficjalnej nazwy i adresu IP zadanego przez nas hosta
4. Następnie za pomocą polecenia `host <nazwa_hosta>` sprawdzamy jaki jest adres IP dla wcześniej sprawdzanej nazwy hosta i porównujemy go z adresem IP, który został zwrócony przez nasz program:  

```
$ host www.o2.pl
```
5. Teraz można posprawdzać adresy IP dla innych nazw oraz zobaczyć jakie komunikaty są wyświetlane przy błędnych nazwach hostów
6. Pytanie sprawdzające:  
Wywołaj nasz program następująco:  

```
$ ./cw1_dns www
```

Następnie zaloguj się na marsa i uruchom program w ten sam sposób, porównaj otrzymane wyniki i spróbuj wyjaśnić dlaczego otrzymamy różne rezultaty skoro w obu przypadkach korzystamy z tych samych serwerów DNS.

## 2.3. Zadanie 2. Uzyskanie nazwy hosta z adresu IP.

Zadanie polega na zmodyfikowaniu gotowego programu `cw1_dns.c`. Zasada działania programu polega na pobraniu jako argumentu wywołania adresu kropkowo-dziesiętnego hosta i wyświetleniu na ekran:

- o Nazwy kanonicznej hosta,
- o Adresu IP hosta.

Przykładowo uruchamiając program z parametrem `212.77.100.101` otrzymamy:

```
Nazwa kanoniczna hosta: www.wp.pl
Adres IP: 212.77.100.101
```

Należy zastosować funkcje `gethostbyaddr`.

## 2.4. Zadanie 3. Uzyskanie adresu IPv6 hosta.

Zadanie polega na napisaniu programu, który po podaniu nazwy kanonicznej jako parametru wywołania wypisze na ekran adres IPv6 hosta.

Po przykładowym wywołaniu programu:

```
$ ./cw3_dns www.ipv6.cbr.tpsa.pl/
```

Na ekranie powinniśmy otrzymać:

```
hostname: www.ipv6.cbr.tpsa.pl
IPv6 address: 3ffe:4019:0:dead:20c:76ff:fe7c:3b12
```

## 2.5. Zadanie 4. Polecenie `tracert`

Poleceniem **tracert** przetestuj łączność z odległym zewnętrznym serwerem. Jako host podaj adres domenowy np. popularnego portalu internetowego.

Sprawdź czy serwer DNS prawidłowo poda numer IP serwera, oraz prześledź drogę pakietu przez serwery pośrednie.

## 2.6. Zadanie 5. Program `wezdate`

Kolejnym zadaniem będzie modyfikacja programu `wezdate.c`. Program ten łączy się z serwerem o podanym adresie ip w celu skorzystania z usługi daytime, o ile taka jest dostępna. Należy zapoznać się z kodem źródłowym programu a następnie sprawdzić jego działanie:

```
kompilacja: gcc wezdate.c -o wezdate
```

```
uruchomienie: ./wezdate <ip serwera>
```

W celu skorzystania z usługi daytime można połączyć się przykładowo z serwerem time.nist.gov. Aby tego dokonać sprawdź jakie ip ma ten serwer używając polecenia nslookup, następnie uruchom program wezdate z odpowiednim parametrem.

Modyfikacja będzie polegała na usprawnieniu działania programu poprzez wprowadzenie możliwości podawania w parametrze nazwy serwera zamiast numeru ip.

Aby odnaleźć adres hosta używamy funkcji:

```
hostinfo = gethostbyname(host);
```

Aby sprawdzić czy serwis jest dostępny wpisujemy:

```
servinfo = getservbyname("daytime", "tcp");
```

Przy okazji uzyskujemy dodatkowe dane takie jak numer portu na którym usługa pracuje.

Powyższe informacje należy przekazać do struktury adresowej gniazda:

```
address.sin_port = servinfo -> s_port;
address.sin_addr = *(struct in_addr *)*hostinfo -> h_addr_list;
```

Po wprowadzeniu powyższych zmian, ponownie skompiluj program i uruchom go, tym razem jako parametr podając nazwę time.nist.gov.

## 2.7. Zadanie 6. Typy rekordów bazy DNS

Użyj wybranego przez siebie sniffera (np.: tcpdump lub ethereal) do prześledzenia sposobu wymiany informacji między maszyną kliencką (np. własny komputer) a serwerem DNS. Użyj polecenia host z argumentem będącym nazwą domenową (np.: host mars.iti.pk.edu.pl) i prześledź jakie rekordy DNS zostały użyte do wymiany informacji w obu kierunkach. Wykonaj także polecenie host z argumentem będącym adresem IP istniejącym w sieci Internet (np.: host 212.77.100.101 - <<jest to odpytanie serwerów wstecznego DNS – ReverseDNS>>) i określ jakie rekordy biorą udział w tej wymianie informacji. Użyj przełączników [-r -v -t] w różnych kombinacjach i określ jakie informacje dostarcza każda z sekcji (QUESTION, AUTHORITY, ADDITIONAL).



### 3. Opracowanie i sprawozdanie

Realizacja laboratorium pt. „Usługa DNS” polega na wykonaniu wszystkich zadań programistycznych podanych w drugiej części tej instrukcji. Wynikiem wykonania powinno być sprawozdanie w formie wydruku papierowego dostarczonego na kolejne zajęcia licząc od daty laboratorium, kiedy zadania zostały zadane.

Sprawozdanie powinno zawierać:

- opis metodyki realizacji zadań (system operacyjny, język programowania, biblioteki, itp.),
- algorytmy wykorzystane w zadaniach (zwłaszcza, jeśli zastosowane zostały rozwiązania nietypowe),
- opisy napisanych programów wraz z opcjami,
- trudniejsze kawałki kodu, opisane tak, jak w niniejszej instrukcji,
- uwagi oceniające ćwiczenie: trudne/łatwe, nie/realizowalne podczas zajęć, nie/wymagające wcześniejszej znajomości zagadnień (wymienić jakich),
- wskazówki dotyczące ewentualnej poprawy instrukcji celem lepszego zrozumienia sensu oraz treści zadań.