

# Sistemul numelor de domenii

**Lenuta Alboaie (adria@info.uaic.ro)**  
**Andrei Panu (andrei.panu@info.uaic.ro)**

# Cuprins

- Domain Name System (DNS)
  - Caracterizare
  - Organizare
  - Configurare
  - Comenzi, Primitive
  - IDN

# DNS

- Adresele IP (ex. 85.122.23.145) sunt dificil de memorat
- Se utilizeaza un **sistem al numelor de domenii** pentru a translata adresele IP in nume de domenii si invers
- Numele de domenii se organizeaza in ierarhii
- RFC 1034, 1035, 1123, 2181

# DNS | organizare

- Initial: **/etc/hosts** – perechi (nume, IP)  
– Probleme de scalabilitate
- Actual: DNS consta dintr-o schema ierarhica de nume de domenii si a unui sistem de baze de date distribuite pentru implementarea acestei scheme de nume

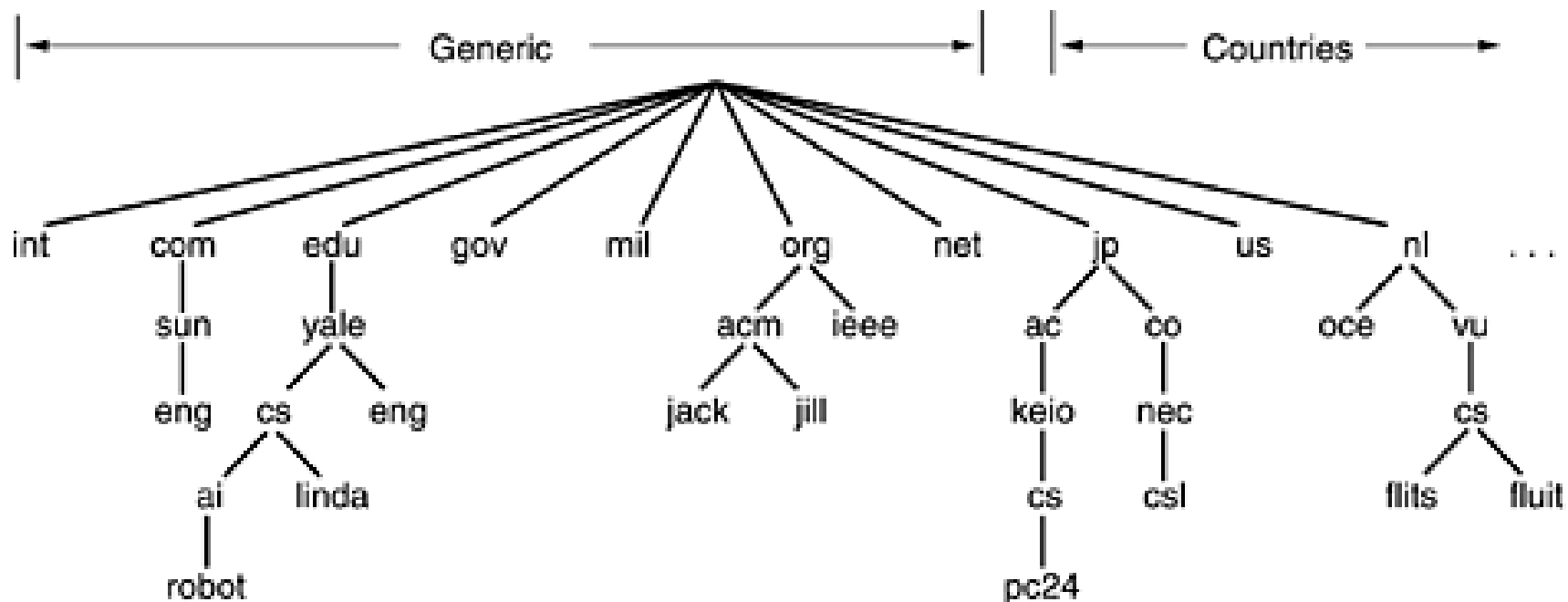


Figura. O portiune a spatiului numelor de domenii in Internet

[Computer Networks, 2003  
Andrew S. Tanenbaum]

# DNS | Tipuri de domenii

- **Primare** (*Top Level Domains* – TLD)
  - pentru Infrastructura Internet – un singur domeniu **.arpa** ARPA (Address and Routing Parameter Area)
    - “Changes to the .arpa zone must be coordinated manually with IANA”
  - State (*ccTLD*) – coduri de state: .ro, .fr, .jp, ...
  - IDN ccTLD (*Internationalized Country Code Top-Level Domains*)  
<http://例子.测试>      <http://example.test>
  - Generice: .biz, .com, .info, .name, .net, .org, .pro
  - Sponsorizate: .aero, .edu, .gov, .int, .jobs, .mil, .tel
  - Rezervate: .example, .invalid, .localhost, .test
  - Pseudo-domenii: .bitnet, .local, .root, .uucp etc.

<http://www.iana.org/domains/root/db/>

# DNS | Tipuri de domenii

## Domain Names

### Overview

### Root Zone Management

Overview

**Root Database**

Hint and Zone Files

Change Requests

Instructions & Guides

Root Servers

.INT Registry

.ARPA Registry

IDN Practices Repository

Root Key Signing Key (DNSSEC)

Reserved Domains

## Root Zone Database

The Root Zone Database represents the delegation details of top-level domains, including gTLDs such as [.com](#), and country-code TLDs such as [.uk](#). As the manager of the DNS root zone, IANA is responsible for coordinating these delegations in accordance with its [policies and procedures](#).

Much of this data is also available via the WHOIS protocol at [whois.iana.org](#).

Domain	Type	Sponsoring Organisation
<a href="#">.abogado</a>	generic	Top Level Domain Holdings Limited
<a href="#">.ac</a>	country-code	Network Information Center (AC Domain Registry) c/o Cable and Wireless (Ascension Island)
<a href="#">.academy</a>	generic	Half Oaks, LLC
<a href="#">.accountants</a>	generic	Knob Town, LLC
<a href="#">.active</a>	generic	The Active Network, Inc
<a href="#">.actor</a>	generic	United TLD Holdco Ltd.
<a href="#">.ad</a>	country-code	Andorra Telecom
<a href="#">.ae</a>	country-code	Telecommunication Regulatory Authority (TRA)
<a href="#">.aero</a>	sponsored	Societe Internationale de Telecommunications Aeronautique (SITA INC USA)
<a href="#">.af</a>	country-code	Ministry of Communications and IT

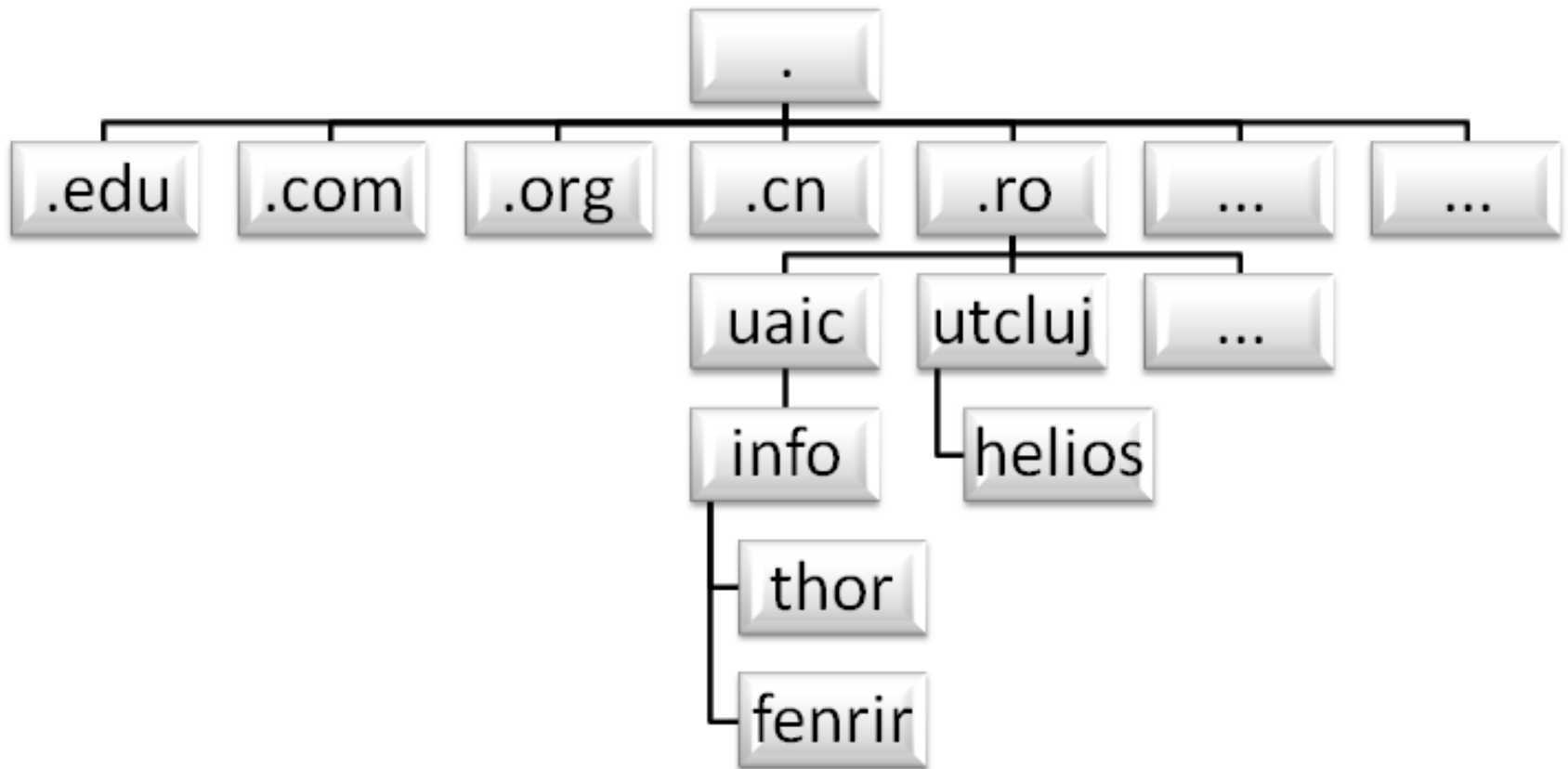
# DNS | Tipuri de domenii

- Domeniu de nume
  - Subarbore al arborelui de domenii
  - Nu trebuie sa respecte topologia rețelei fizice
- **Sub-domenii:**
  - întreaga cale de nume nu depășește 255 de caractere
- **Nume de calculatoare (gazde)**



# DNS

- Exemplu:



# DNS | organizare

- Reguli de alocare a numelor de domenii:
  - Fiecare domeniu controleaza cum sunt alocate subdomeniile sale
  - Pentru a crea un nou subdomeniu, se cere permisiunea domeniului in care va fi inclus (un domeniu de la un anumit nivel va avea o autoritate)
  - Atribuirea de nume de domenii respecta granitele organizationale, nu pe cele ale retelelor
  - Un anumit nivel din ierarhia de niveluri poate fi controlat de mai multe servere

# DNS | organizare

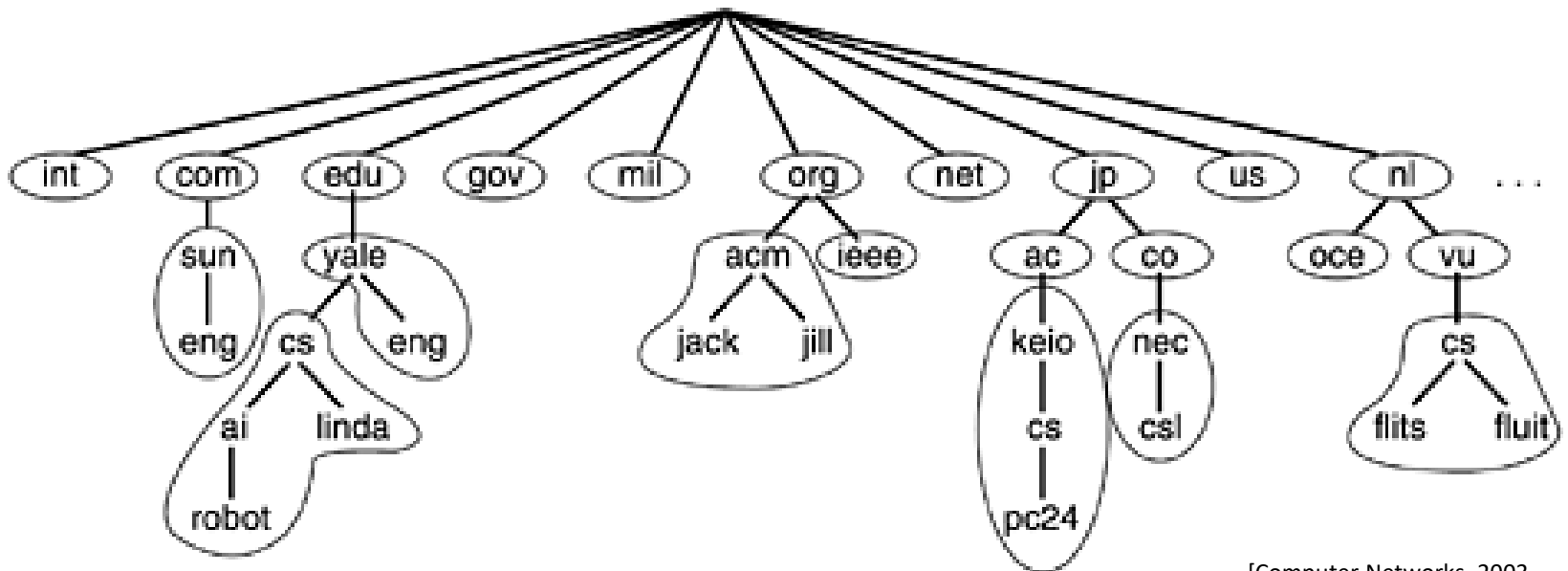
- Servere de nume (*name servers*)
  - Teoretic, un singur server de nume poate contine intreaga baza de date DNS si poate raspunde tuturor cererilor
    - Probleme: incarcarea si “*single point of failure*”
- Spatiul de nume DNS se divide in zone ne-suprapuse



# DNS | organizare

- Servere de nume (name servers)

Exemplu: O posibila impartire a spatiului de nume DNS in zone



[Computer Networks, 2003  
Andrew S. Tanenbaum]

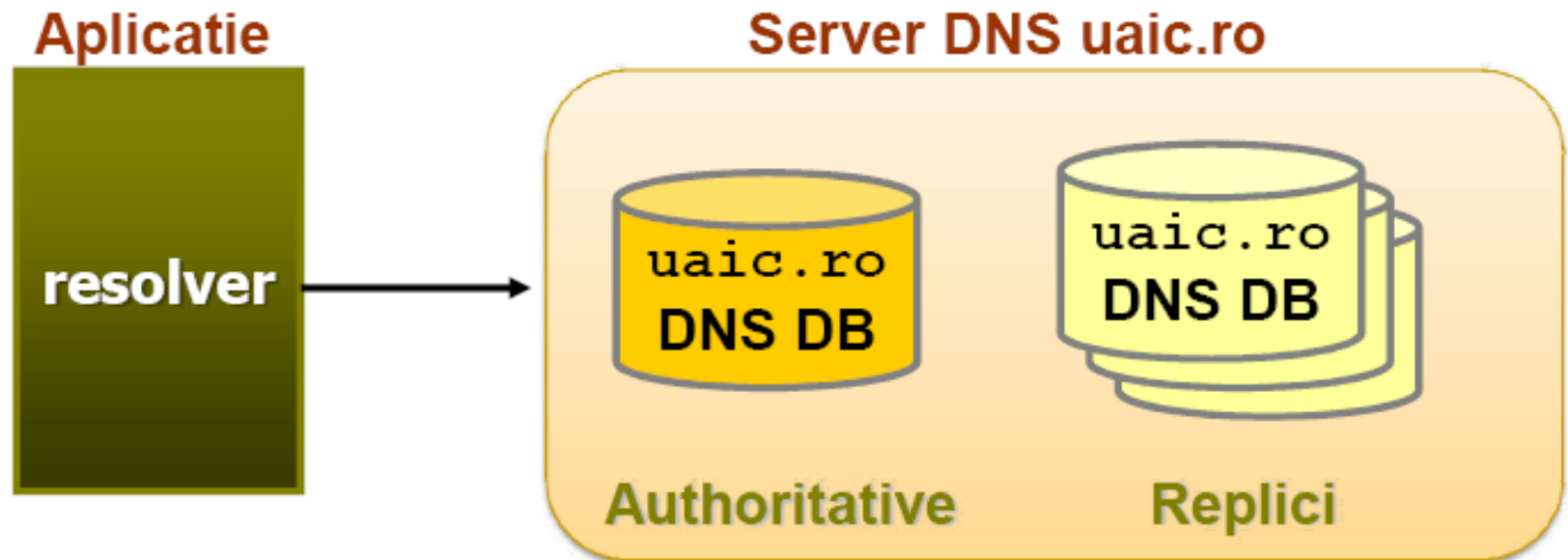
12

# DNS | organizare

- Servere de nume (name servers)
  - Exista un **server primar** (*primary/authoritative name server*) care deserveste un anumit domeniu si, eventual, mai multe **servere secundare** continind baze de date replicate
  - TCP se utilizeaza pentru replicarea DNS
  - UDP pentru interogari (*lookups*)

# DNS | organizare

- Client DNS
  - Denumit **resolver**, trimite un pachet UDP serverului DNS care cauta numele si returneaza adresa IP



# DNS | organizare

- Exemplu de implementari ale serverului de nume: **BIND** (Berkeley Internet Name Domain), MSDNS, PowerDNS etc.
- Ca resolver (client) interactiv, se poate folosi de exemplu una dintre comenzile: **nslookup**, **host** sau **dig**.

# DNS | interogari

- Interogari:
  - **Recursiva** – daca un server DNS nu cunoaste adresa pentru numele solicitat, atunci va interoga alt server DNS
  - **Incrementala** – daca serverul DNS nu stie sa raspunda, returneaza eroare si adresa altui server DNS (numit si *referral*) care ar putea cunoaste raspunsul la interogare

[<http://technet.microsoft.com/en-us/library/cc775637%28v=ws.10%29.aspx>]



# DNS | interogari

- Fiecarui domeniu ii este asociata o multime de inregistrari de resurse (**resource record – RR**)
- Mecanismul:
  - Cererea: *resolver*-ul trimite un nume de domeniu
  - Raspunsul: inregistrarile de resurse asociate acelui nume (stocate in bazele de date DNS)



DNS realizeaza corespondenta dintre numele de domenii si inregistrarile de resurse

# DNS | interogari

- Forma generala RR este:

Nume\_domeniu Timp\_de\_viata Tip Clasa Valoare

Nume\_domeniu (*domain name*) – precizeaza domeniul caruia i se aplica aceasta inregistrare

Timp\_de\_viata (time-to-live) – da o indicatie despre cat de stabila este inregistrarea

# DNS | interogari

## Tip - precizeaza tipul inregistrarii

- **SOA** (*Start Of Authority*) : domeniul curent, adresa e-mail a administratorului, etc.
- **A** – adresa IP a gazdei
- **MX** (mail exchangers) – precizeaza numele domeniului pregatit sa accepte posta electronica pentru domeniul specificat
- **CNAME** (*Canonical Name*) – permite crearea pseudonimelor
- **PTR** (Pointer) – Pseudonim pentru adresa IP
- **HINFO**- permit aflarea tipului de masina si de sistem de operare carora le corespunde domeniul
- **TXT**: text neinterpretat (comentarii)

# DNS | interogari

**Clasa:** pentru Internet valoarea acestuia este IN

**Valoare:** acest camp poate fi un numar, un nume de domeniu sau un sir ASCII; semantica depinde de tipul de inregistrare

Exemple de  
inregistrari  
de  
resurse DNS

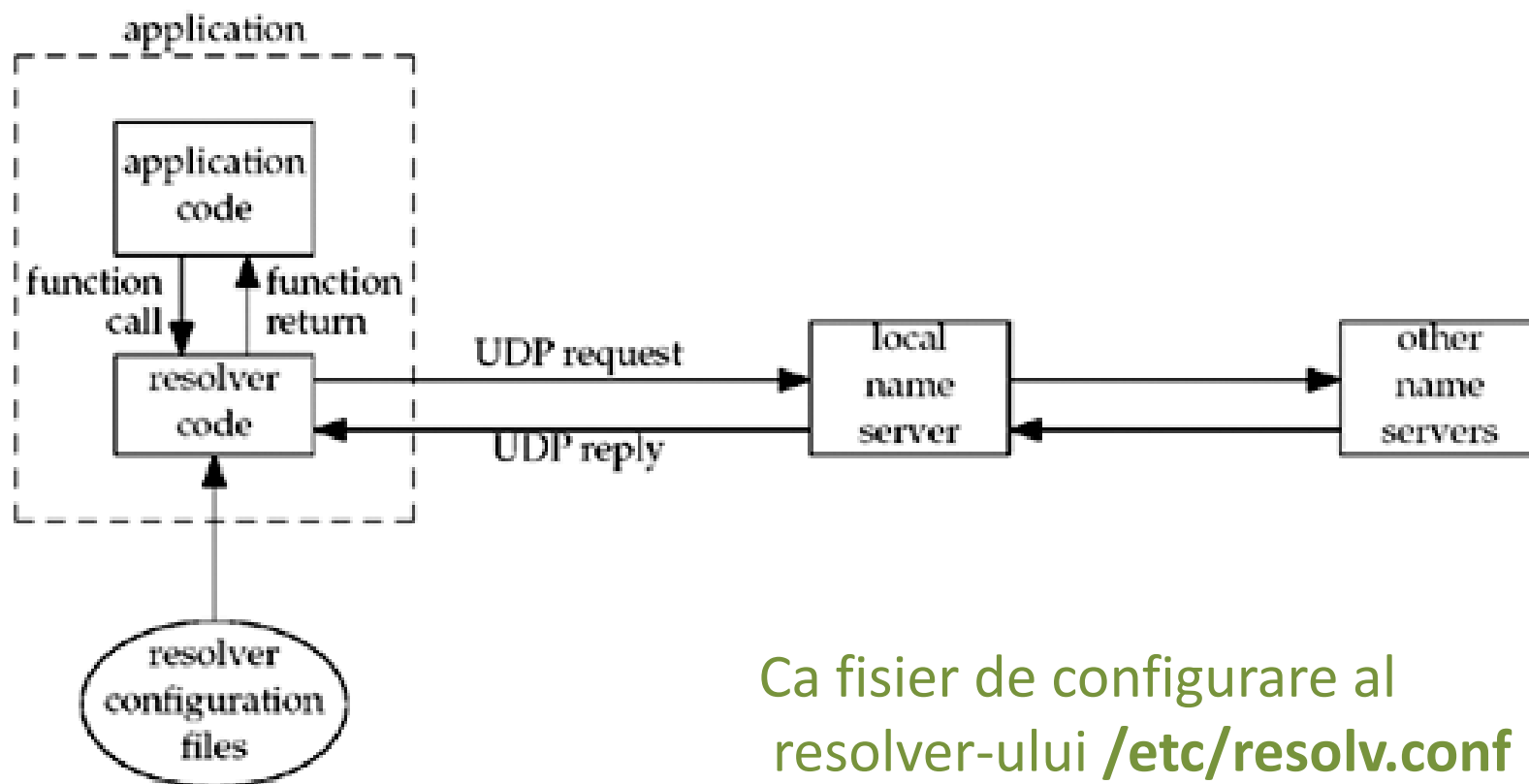
Type	Meaning	Value
SOA	Start of Authority	Parameters for this zone
A	IP address of a host	32-Bit integer
MX	Mail exchange	Priority, domain willing to accept e-mail
NS	Name Server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
HINFO	Host description	CPU and OS in ASCII
TXT	Text	Uninterpreted ASCII text

# DNS | configurare

- Exemplu de fisier pentru specificarea unei zone DNS

```
; Zone file for axiologic.ro
;
; The full zone file
;
$TTL 3D
@      IN      SOA      ns1.axiologic.ro. abss.axiologic.ro. (
                        2007050103      ; serial, todays date + todays serial #
                        14400           ; refresh, seconds
                        7200            ; retry, seconds
                        1209600         ; expire, seconds
                        1D )            ; minimum, seconds
;
@      NS      ns1.axiologic.ro.      ; Inet Address of name server
@      NS      ns2.axiologic.ro.      ; Inet Address of name server
@      MX      5 mailx.axiologic.ro.   ; Primary Mail Exchanger
;
localhost      A      127.0.0.1
axiologic.ro.  A      72.249.105.153
www            A      72.249.105.153
mailx          CNAME  axiologic.net.
mail           A      207.210.101.144
ftp            A      72.249.105.153
axiologic.ro. IN TXT  "v=spf1 mx mx:mailx.axiologic.ro. ~all"
ns1            A      207.210.101.144
ns2            A      207.210.101.216
~
(END)
```

# DNS| clienti, resolveri, servere



Ca fisier de configurare al  
resolver-ului **/etc/resolv.conf**

[Unix Network Programming, R. Stevens  
B. Fenner, A. Rudoff - 2003]

# DNS | configurare

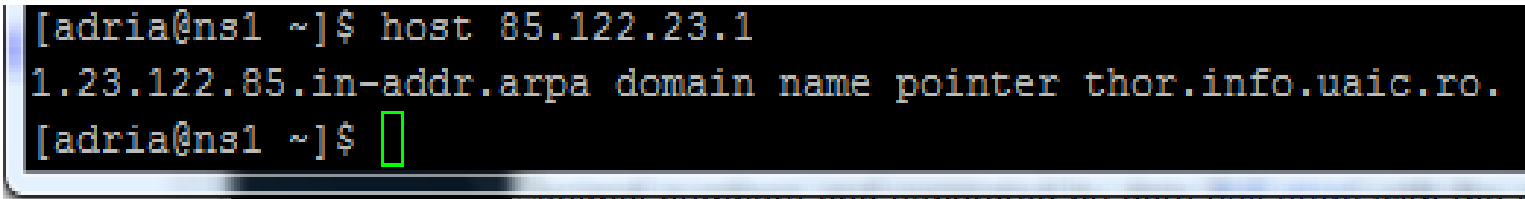
- Exemplu de fisier /etc/resolv.conf

```
[adria@thor ~] $ cat /etc/resolv.conf
domain info.uaic.ro
search info.uaic.ro
nameserver 85.122.16.1
nameserver 85.122.16.4
[adria@thor ~] $
```

# DNS | interogari inverse

- Problema:
  - Data o adresa, care va fi numele ei simbolic?  
(*reverse DNS resolution* sau *reverse DNS lookup*)

Exemple:

1) A terminal window with a black background and white text. The prompt is [adria@ns1 ~]\$. The command 'host 85.122.23.1' is entered. The output is '1.23.122.85.in-addr.arpa domain name pointer thor.info.uaic.ro.'. The prompt is followed by a green cursor.

2) 2001:db8::567:89ab  
b.a.9.8.7.6.5.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa



# DNS | optimizari

**Proximitatea spatiala:** serverele locale vor fi interogate mai des decat cele la distanta

**Proximitatea temporală:** daca un set de domenii sunt referentiate repetat atunci se apeleaza la *caching DNS*

Fiecare intrare DNS va avea stabilita o valoare TTL (*time to live*)

Se va utiliza si replicarea (servere multiple, servere root multiple) – se va interoga cel mai apropiat (geografic) server

# DNS | comenzi

Ca resolver interactiv se pot folosi comenzile:

- **nslookup**
- **dig**
- **host**
- **whois**
- ...

# DNS | nslookup

## Exemple de utilizari:

➤ **nslookup** www.info.uaic.ro

- Returneaza RR de tip A folosind serverul DNS local

```
[adria@thor ~] $ nslookup www.info.uaic.ro
Server:          85.122.16.1
Address:         85.122.16.1#53

www.info.uaic.ro      canonical name = vidar.info.uaic.ro.
Name:   vidar.info.uaic.ro
Address: 85.122.23.146
```

*Host Lookup*

➤ **nslookup** 85.122.23.1

- Returneaza RR de tip PTR pentru 85.122.23.1 in ierarhia de domenii in-addr.arpa

```
[adria@thor ~] $ nslookup 85.122.23.1
Server:          85.122.16.1
Address:         85.122.16.1#53

1.23.122.85.in-addr.arpa      name = thor.info.uaic.ro.
```

*Reverse IP  
Lookup*

[<http://www.zytrax.com/books/dns/ch3/>]

# DNS | nslookup

## Exemple de utilizari:

➤ **nslookup** www.axiologic.ro

- Returneaza RR de tip A folosind serverul DNS specificat

```
adria@thor:~$ nslookup www.axiologic.ro 207.210.101.144
Server:          207.210.101.144
Address:         207.210.101.144#53

Name:   www.axiologic.ro
Address: 72.249.105.153
```

*Host Lookup*

➤ **man nslookup**

# DNS| dig

**dig** – un instrument mai puternic decat nslookup

Exemplu de  
utilizare:

```
[adria@thor ~] $ dig www.info.uaic.ro A

; <<>> DiG 9.6-ESV-R4 <<>> www.info.uaic.ro A
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19336
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 4

;; QUESTION SECTION:
;www.info.uaic.ro.                IN      A

;; ANSWER SECTION:
www.info.uaic.ro.                86400   IN      CNAME   vidar.info.uaic.ro.
vidar.info.uaic.ro.             86400   IN      A       85.122.23.146

;; AUTHORITY SECTION:
info.uaic.ro.                   86400   IN      NS       orion.uaic.ro.
info.uaic.ro.                   86400   IN      NS       onix.uaic.ro.
info.uaic.ro.                   86400   IN      NS       ns.iasi.roedu.net.

;; ADDITIONAL SECTION:
ns.iasi.roedu.net.              86400   IN      A       192.129.4.100
ns.iasi.roedu.net.              86400   IN      AAAA    2001:b30:1:100::100
onix.uaic.ro.                   86400   IN      A       85.122.16.4
orion.uaic.ro.                  86400   IN      A       85.122.16.1

;; Query time: 1 msec
;; SERVER: 85.122.16.1#53(85.122.16.1)
;; WHEN: Mon Nov 14 11:57:27 2011
;; MSG SIZE rcvd: 216
```

➤ **dig** www.info.uaic.ro A

# DNS | comenzi

## host

### Exemplu de utilizare:

```
adria@thor:~$ host 128.30.52.45
45.52.30.128.in-addr.arpa domain name pointer dolph.w3.org.
```

# DNS| whois

whois ibm.com

```
Registrar:
International Business Machines Corporation
New Orchard Road
Armonk, NY 10504
US

Domain Name: IBM.COM

-----
Promote your business to millions of viewers for only $1 a month
Learn how you can get an Enhanced Business Listing here for your domain name
Learn more at http://www.NetworkSolutions.com/
-----

Administrative Contact:
IBM DNS Admin                      dnsadm@us.ibm.com
IBM Corporation
New Orchard Road
Armonk, NY 10504
US
+1.9147654227 fax: +1.9147654370

Technical Contact:
IBM Corporation                      ipreg@us.ibm.com
New Orchard Road
Armonk, NY 10504
US
+1.9192544441 fax: +1.9147654370

Record expires on 20-Mar-2018.
Record created on 19-Mar-1986.
Database last updated on 8-Nov-2010 04:12:22 EST.

Domain servers in listed order:

INTERNET-SERVER.ZURICH.IBM.COM 195.176.20.204
NS.WATSON.IBM.COM               129.34.20.80
NS.ALMADEN.IBM.COM              198.4.83.35
NS.AUSTIN.IBM.COM               192.35.232.34

adria@thor:~$
```

# DNS | primitive

- Nu trebuie scris un resolver pentru a afla adresa IP a unei gazde
- Functii principale:
  - `gethostbyname(); getaddrinfo();`
  - `gethostbyaddr() ; getnameinfo();`
- La unele sisteme de operare (e.g., Solaris) va trebui la compilare sa folosim biblioteca *ns* (*Name Server Library*): `gcc ... -lnsl`



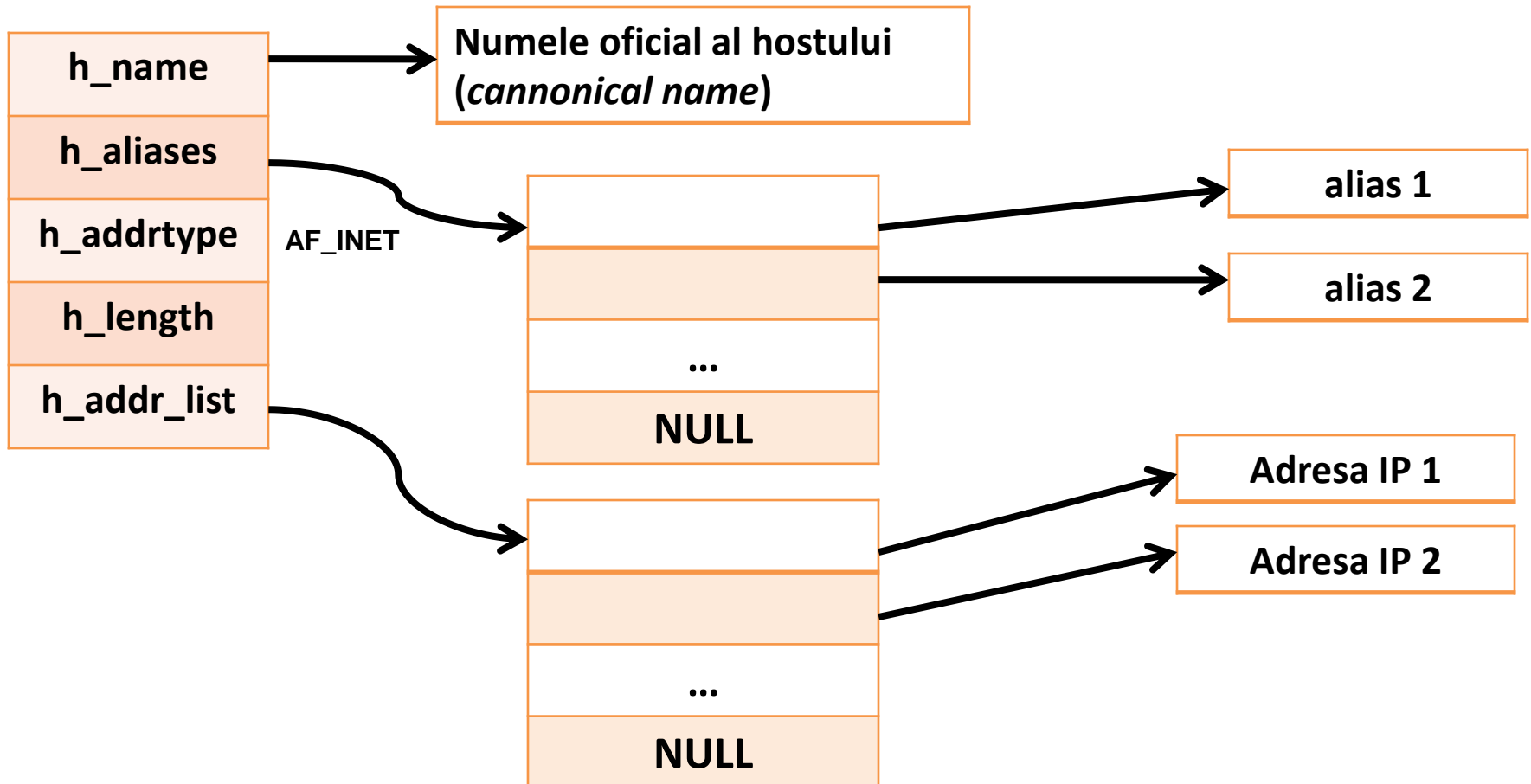
# DNS| primitive

Una din structurile folosite: **hostent**

```
struct hostent {  
    char *h_name;      /* nume oficial (canonical) */  
    char **h_aliases; /* alias-uri */  
    int h_addrtype;    /* AF_INET */  
    int h_length;      /* lungimea adresei: 4 sau 6 */  
    char **h_addr_list; /* pointeri la adresele IP */  
};
```

# DNS | primitive

## Structura **hostent**:




# DNS | gethostbyname()

```
#include <netdb.h>
```

```
struct hostent *gethostbyname  
                (const char *hostname);
```

- In termenii DNS, `gethostbyname()` realizeaza o cerere pentru o inregistrare **A**
- Obs. `gethostbyname()` se foloseste in special pentru IPv4

# DNS | gethostbyname()

- Returneaza:
    - In caz de succes returneaza un pointer la **hostent**, ce contine adresa IP a *host*-ului
    - In caz de eroare NULL, iar variabila **h\_errno** indica eroarea aparuta:
      - **HOST\_NOT\_FOUND**
      - ...
      - **NO\_RECOVERY**
      - ...
- 

Constante definite in  
**netdb.h**

# DNS | gethostbyname()

- Exemplu de utilizare: completarea structurii *sockaddr\_in* avind in loc de adresa IP un nume simbolic:

```
struct sockaddr_in server;  
struct hostent *hos;  
if(! ( hos = gethostbyname("fenrir.info.uaic.ro") )  
    { /* Eroare la rezolvarea adresei */ }  
server.sin_family=AF_INET  
    /* adresa IP o luam din structura hos */  
memcpy(&server.sin_addr.s_addr, hos->h_addr_list[0],  
        sizeof(hos->h_addr_list));  
server.sin_port=htons(4321);
```

# DNS | gethostbyaddr()

```
#include <netdb.h>
```

```
struct hostent *gethostbyaddr (  
    const char *addr,  
    socklen_t len,  
    int family);
```

- In termenii DNS, **gethostbyaddr()** realizeaza o cerere la serverul de nume pentru o inregistrare **PTR** in domeniul in-addr.arpa
- Returneaza: In caz de succes returneaza un pointer la **hostent**, ce contine numele oficial al *host*-ului ; In caz de eroare NULL, iar variabila **h\_errno** indica eroarea aparuta

Obs. **gethostbyaddr()** se foloseste in special pentru IPv4

# DNS | getservbyname()

```
#include <netdb.h>
```

```
struct servent *getservbyname (const char *servname, const char  
    *protoname);
```

- Returneaza: un pointer la *struct servent* in caz de succes, NULL in caz de eroare

```
struct servent {  
    char *s_name;    /* numele oficial al serviciului */  
    char **s_aliases; /* alias-uri */  
    int s_port;      /* portul (network-byte order) */  
    char *s_proto;  /* protocolul */ };
```

Exemplu: struct servent \*pserv;

```
pserv=getservbyname("ftp","tcp"); /*FTP folosind TCP */
```

39

# DNS | getservbyport()

```
#include <netdb.h>
```

```
struct servent *getservbyport (int port, const char *protoname);
```

- Cauta un serviciu dupa un numar de port si dupa protocol (optional)
- Returneaza: un pointer la *struct **servent*** in caz de succes, NULL in caz de eroare

Obs. port este in *network byte order*

## Exemplu:

```
struct servent *pserv;
```

```
pserv=getservbyport( htons(53), "udp"); /*DNS folosind UDP */
```

```
pserv=getservbyport( htons(21),"tcp"); /*FTP folosind TCP */
```



# DNS | getaddrinfo()

```
#include <netdb.h>
```

```
int getaddrinfo (  
    const char *hostname,  
    const char *service,  
    const struct addrinfo *hints,  
    struct addrinfo **result ) ;
```

Numele host-lui sau o adresa IPv4 sau IPv6 ca string

Portul serviciului sau numele serviciului ("http","pop",...) (vezi /etc/services )

Contine informatii despre tipul de informatii pe care trebuie sa le intoarca primitiva

- Obs. *hostname*, *service*, *hints* – parametri de intrare
- Returneaza: 0 in caz de succes, !=0 in caz de eroare
- Se recomanda a fi folosita si pentru IPv4 si pentru IPv6
- Combina functionalitati ale: `gethostbyname()`, `getservbyname()`, `getservbyport()`

# DNS | getaddrinfo()

```
struct addrinfo {  
    int ai_flags;          /* AI_PASSIVE, AI_CANONNAME */  
    int ai_family;        /* AF_INET, AF_INET6, AF_UNSPEC */  
    int ai_socktype;      /* SOCK_STREAM sau SOCK_DGRAM */  
    int ai_protocol;     /* 0 (auto) sau IPPROTO_TCP, IPPROTO_UDP */  
    socklen_t ai_addrlen; /* lungimea lui ai_addr */  
    char *ai_canonname;   /* numele canonic al host-ului */  
    struct sockaddr *ai_addr; /* adresa binara a socket-ului */  
    struct addrinfo *ai_next; /* pointer la urmatoarea structura din  
    lista */  
};
```

# DNS | getaddrinfo()

Discutii:

- Daca functia returneaza cu succes **result** va pointa la lista de **struct addrinfo**.

Cazuri cind se pot obtine structuri multiple:

- Exista mai multe adrese asociate cu numele hostului, si cate o structura este returnata pentru fiecare adresa
- Daca serviciul este furnizat pentru tipuri diferite de *socket*-uri, atunci cate o structura este returnata pentru fiecare tip de *socket*
- Informatia returnata de **getaddrinfo()** in structura **struct addrinfo** **\*\*result** poate fi utilizata astfel:
  - Pentru **socket()** : **ai\_family**, **ai\_socktype**, **ai\_protocol**
  - Pentru **connect()** sau **bind()**: **ai\_addr** si **ai\_addrlen**
- **freeaddrinfo()**

# DNS | getnameinfo()

```
#include <netdb.h>
```

```
int getnameinfo (  
    const struct sockaddr *sockaddr,  
    socklen_t addrlen,  
    char *host,  
    socklen_t hostlen,  
    char *serv,  
    socklen_t servlen,  
    int flags) ;
```

Adresa socket-ului trimisa ca argument

numele host-ului intors

Numele serviciului

NI\_NOFQDN -> **host** va contine doar numele host-ului si nu intreg numele al domeniului

- Inlocuieste gethostbyaddr() si getservbyport()
- Returneaza: 0 in caz de succes, !=0 in caz de eroare

# DNS | IDN

- **International Domain Names (IDN)**

- Extensie care permite folosirea caracterelor Unicode in numele de domenii, nu doar a celor ASCII

<http://www.icann.org/en/topics/idn/>

**16 Noiembrie 2009** - Inregistrarea de domenii ccIDN sau IDN ccTLD

2010-01: ICANN announces that Egypt, the Russian Federation, Saudi Arabia, and the United Arab Emirates were the first countries to have passed the Fast Track String Evaluation within the IDN ccTLD domain application process.

- Pot fi exploatare pentru atacuri de tip *phishing*  
(... detalii intr-un curs viitor)

# DNS | administrare

- Radacina DNS este oficial administrata de Internet Corporation for Assigned Names and Numbers (ICANN)
- Exista si alte organizatii care ofera radacini alternative (alt DNS roots), precum OpenNIC (Network Information Center) sau New.Net

# Rezumat

- Domain Name System (DNS)
  - Caracterizare
  - Organizare
  - Configurare
  - Comenzi, Primitive
  - IDN



# Intrebari?

Intrebari?