Sistemul numelor de domenii (II)

Lenuta Alboaie
adria@infoiasi.ro

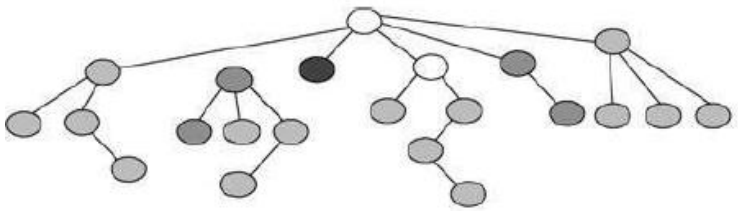
Cuprins

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

DNS| … sa ne reamintim

- **DNS** – poate fi privit ca o baza de date distribuita utilizata la maparea dintre numele host-urilor si IP-uri si *vice versa*

- **DNS - Structura**



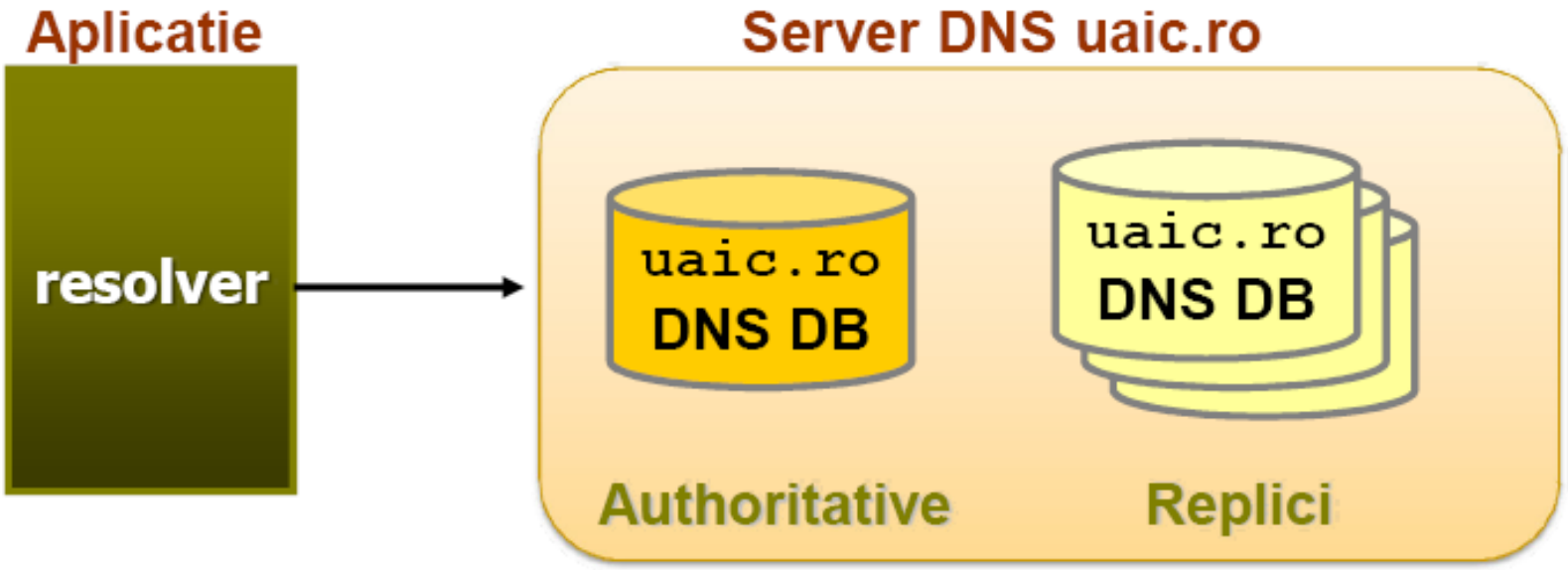
- structura ierarhica
  - domenii
  - subdomenii
  - … (curs anterior)

DNS| … sa ne reamintim

- **DNS – componente**
  - spatiul numelor de domenii si RRs (resource records)
    - Spatiul numelor de domenii este impartit in zone nesuprapuse (zone)
    - RR – inregistrari in baza de date DNS
      - Tipuri:
        - » **A** – adresa IP a gazdei
        - » …(vezi cursul anterior)
  - servere de nume (name servers)
    - *root name servers*
    - server primar (*primary/authoritative name server)*
    - servere secundare
  - clienti DNS sau resolvers

DNS| … sa ne reamintim

- Client DNS *(resolver)*, trimite un pachet UDP serverului DNS care cauta numele si returneaza adresa IP sau invers



[Retele de calculatoare – curs 2007-2008, Sabin Buraga]

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:

- **dig www.info.uaic.ro A**

```
[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
```

DNS| comenzi

host

Exemplu de utilizare:

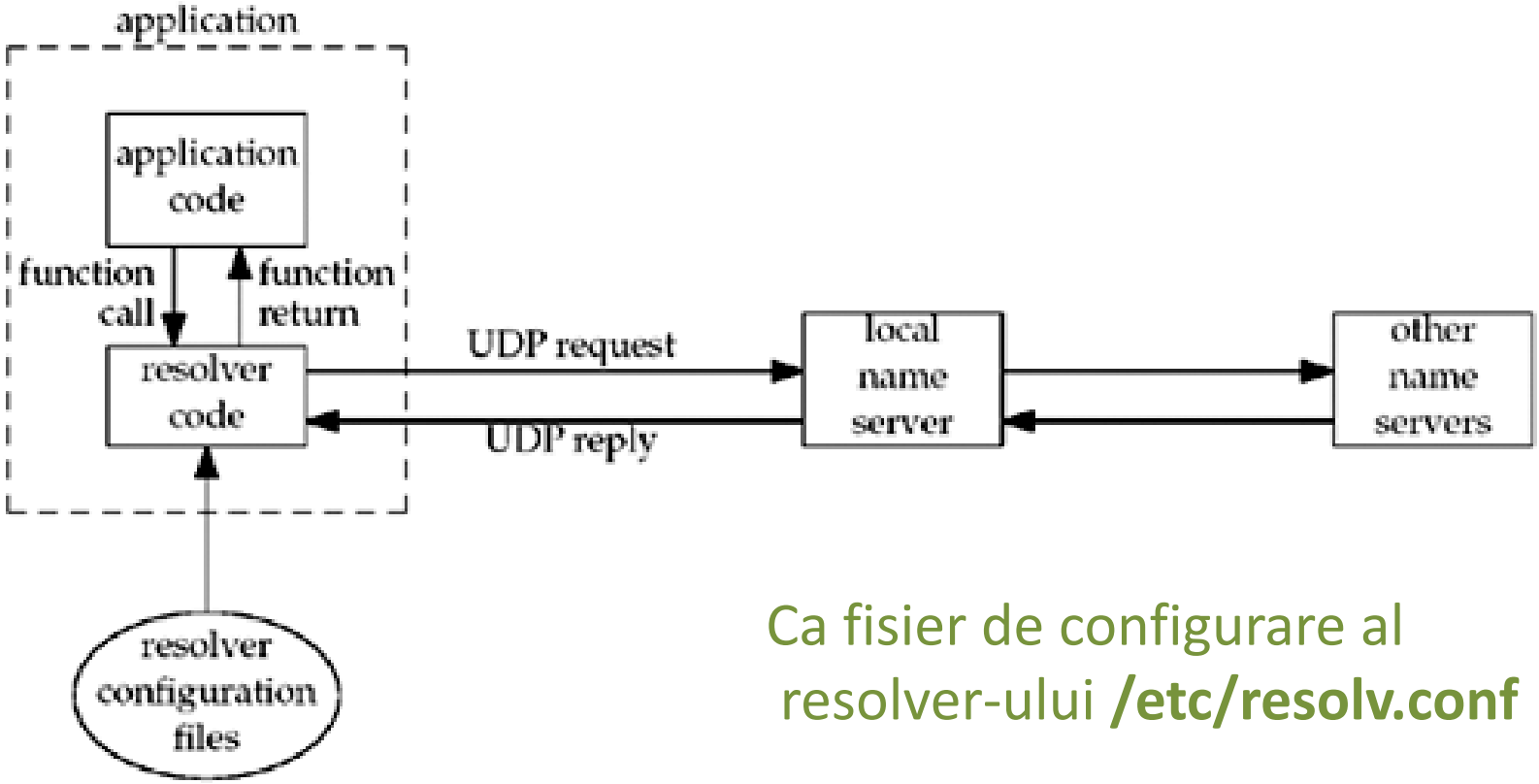
```
adria@thor:~$ host 128.30.52.45
45.52.30.128.in-addr.arpa domain name pointer dolph.w3.org.
```

DNS| whois

whois ibm.com

```
Registrant:
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 nam
   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| clienti, resolveri, servere



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

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

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 nsl (*Name Server Library*): gcc … -lnsl

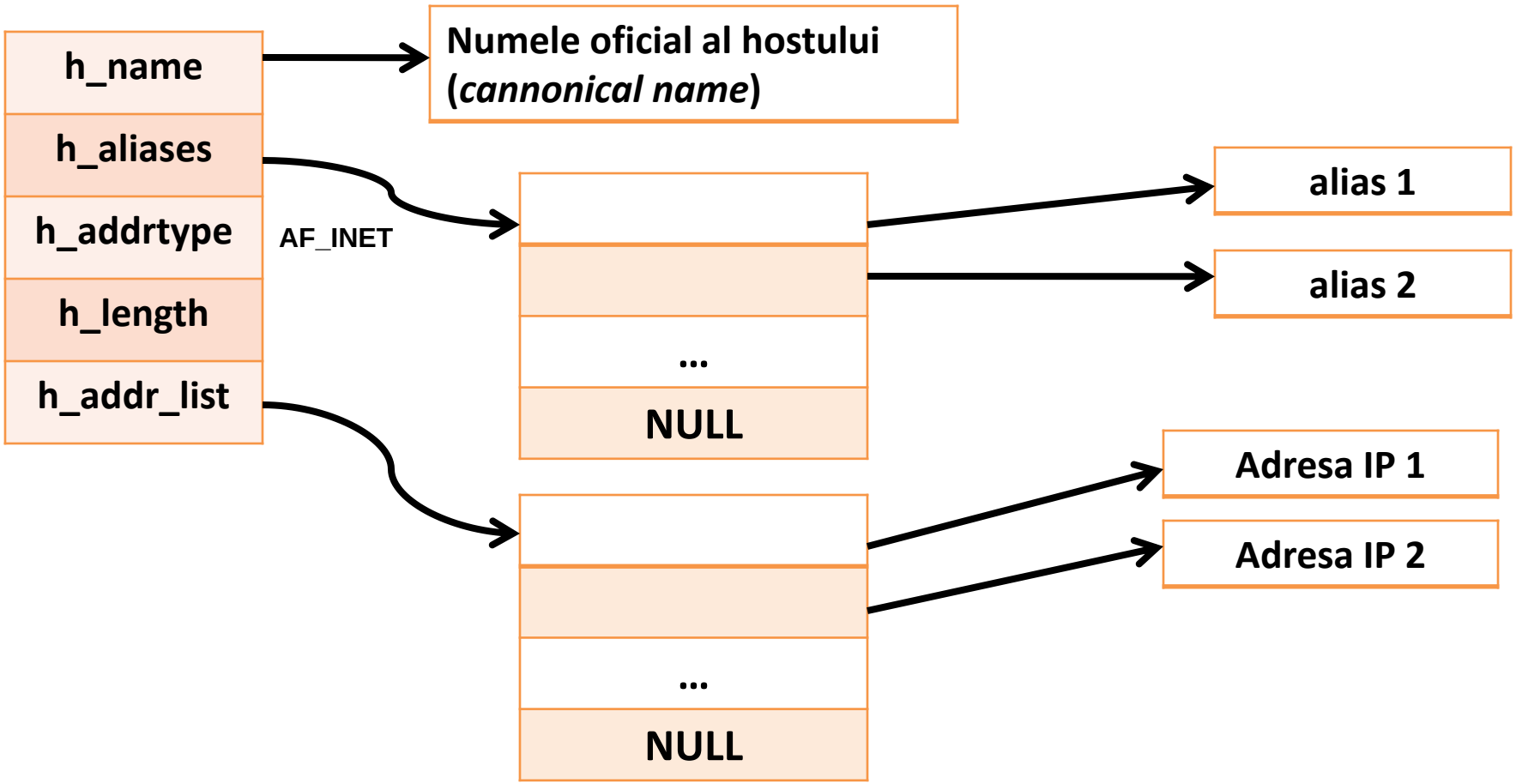
DNS| primitive

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:**



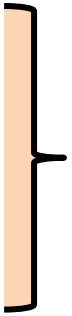
DNS| gethostbyname()

```
#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
    - NO_ADDRESS



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 sucess, 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 */
```

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 sucess, 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 sucess, !=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 *soket*-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 soketu-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 sucess, !=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 exploatate 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?