



Reti di Calcolatori - Laboratorio

Lezione 3

Gennaro Oliva



Indirizzi IP

- Fino ad ora abbiamo indentificato ogni host mediante un indirizzo IP
Es. 192.167.11.34
- Sarebbe opportuno poter utilizzare dei nomi simbolici più semplici da ricordare
- Con IPV6 gli indirizzi diventeranno particolarmente lunghi
(128 bit contro gli attuali 32)

Fully Qualifeyed Domain Name

- Sulla rete Internet gli host vengono identificati mediante un nome simbolico detto fully qualified domain name (FQDN)
- Il FQDN è composto dal nome locale dell'**host** e dal **dominio di appartenenza**
ventotene.dma.unina.it
- L'associazione tra nomi simbolici degli host ed i corrispondenti indirizzi viene effettuata dal Domain Name System

Domain Name System

- Il Domain Name System e' un database distribuito che memorizza le altre le seguenti tipologie di record:
 - A che associano FQDN ad indirizzi IPv4
 - AAAA che associano FQDN ad indirizzi IPv6
 - PTR che associano un indirizzo IP ad un hostname
 - MX che specificano chi agisce da mail exchanger per un determinato dominio
 - CNAME che forniscono alias per un host

Server DNS

- L'operazione di trovare l'indirizzo IP associato ad un hostname viene detta risoluzione diretta
- L'operazione contraria e' detta risoluzione inversa
- La risoluzione sulla rete internet viene effettuata effettuando query ad un server DNS

Server DNS

- Dalla shell è possibile utilizzare i comandi nslookup, host e dig
- Il comando host ci permette di effettuare query al DNS

```
host informatica.uninav.it
```

```
host 192.167.9.15
```

```
host -t AAAA www.ipv6.org
```

```
host -t MX unina.it
```

resolver

- Il linguaggio C fornisce un insieme di routine detto resolver che consentono di effettuare query al DNS dalle applicazioni scritte in C
- Il DNS non è l'unico servizio a fornire associazione tra indirizzi e host
- Alternative al DNS sono il file locale `/etc/hosts` o servizi come NIS o LDAP

Alternative al DNS

- Il file `/etc/hosts` e' un file di testo che associa indirizzi IP e hostname le cui linee hanno formato:

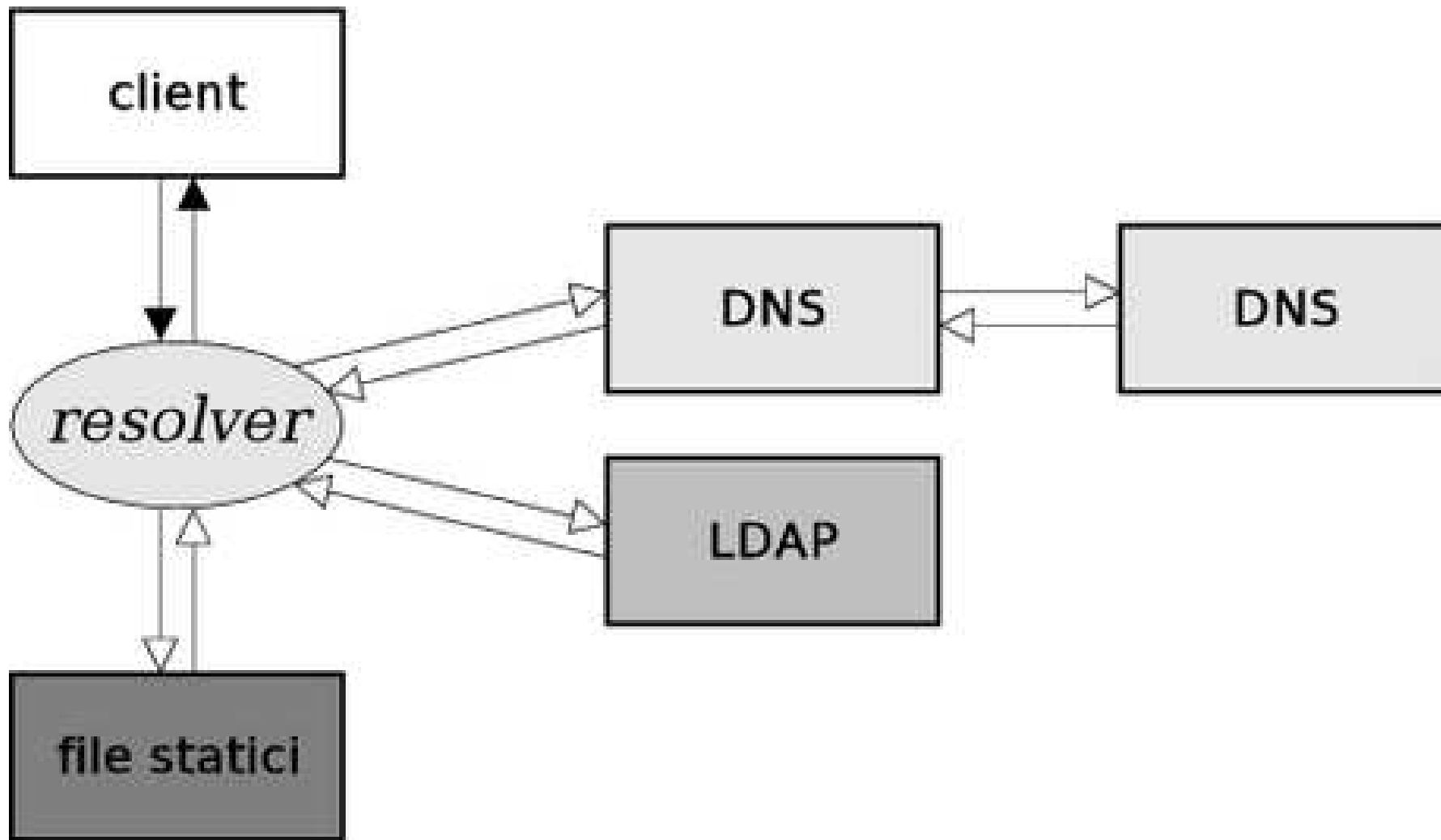
`indirizzo_IP hostname_canonico [alias...]`

- Di regola il file `/etc/hosts` viene consultato prima di interrogare il DNS per cui si e' solito inserirvi gli host acceduti piu' frequentemente

Alternative al DNS

- L'ordine con cui il resolver utilizza le alternative possibili e' stabilito dal system administrator
- Nei sistemi Linux questa informazione e' conservata nel file `/etc/nsswitch.conf`
- Il nome del server DNS e' contenuto nel file `/etc/resolv.conf`

Struttura del resolver



gethostbyname

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

- Restituisce una struttura di tipo hostent contenente varie informazioni sul host passato come argomento, NULL in caso di errore
- Accetta come argomento sia un nome simbolico che un indirizzo

gethostbyname

```
struct hostent {
```

stringa zero-terminata

```
char *h_name; /* nome canonico dell'host */
```

array di string il cui ultimo elemento è null

```
char **h_aliases; /* lista di alias */
```

lunghezza degli indirizzi

```
int h_addrtype; /* famiglia dell'indirizzo */
```

famiglia degli indirizzi

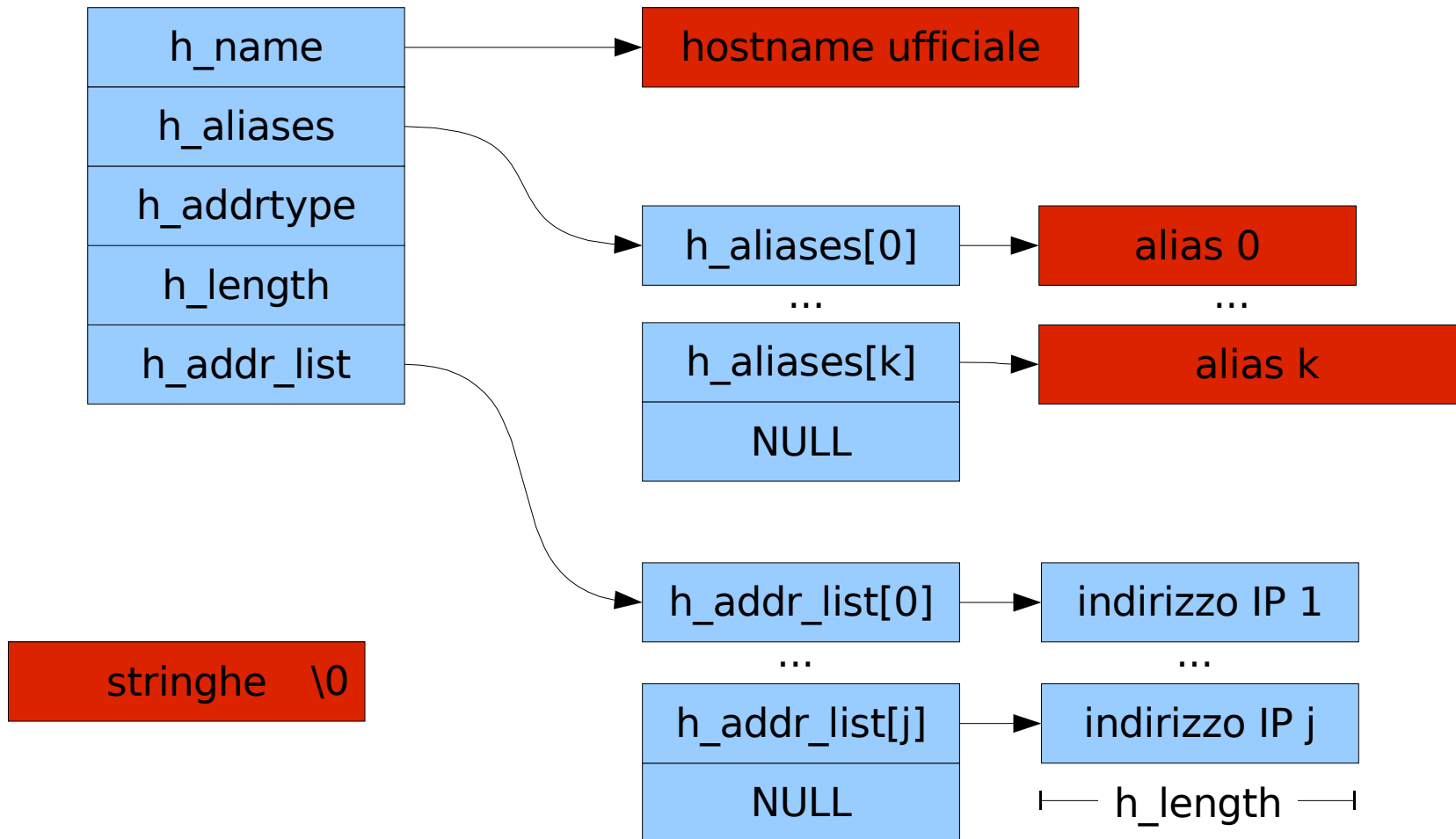
```
int h_length; /* lunghezza dell'indirizzo */
```

lista di indirizzi IP dell'host

```
char **h_addr_list; /* lista di indirizzi */
```

```
}
```

hostent



inet_ntop

```
const char *inet_ntop(int f, const void *src,  
                      char *dst, socklen_t cnt);
```

- Converte l'indirizzo memorizzato in formato network in src in notazione dotted (192.167.11.34) memorizzandolo in dst
- f e' la famiglia dell'indirizzo
- cnt e' la lunghezza di src

gethostbyaddr

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

- La funzione inversa di gethostbyname
- Restituisce un puntatore ad una struttura hostent che corrisponde all'indirizzo passato con primo argomento addr, NULL in caso di errore

gethostbyaddr

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

- addr non e' realmente un puntatore a char ma e' un puntatore ad una struttura in_addr

```
struct in_addr {  
    unsigned long s_addr; // load with inet_aton()  
};
```

- Il resolver interroga il DNS per i record PTR

herror

```
void herror(const char *s);
```

- Sostituisce perror in caso di errore delle funzioni del resolver

Esercizio 1

- Modificare il client daytime in modo che accetti anche nomi simbolici come parametro
- Modificare il server daytime in modo che visualizzi sullo standard output l'hostname dei client che lo contattano
- Esercizio3: Scrivere un server concorrente che accetti dai client stringhe di caratteri e restituisca il numero di caratteri in esse contenute