

Adresconversies en DNS

Veerle Ongenae

DNS

- Omzetting DNS-namen $\leftrightarrow$ IP
- DNS in C/C++ -programma
 - Niet expliciet DNS-servers aanspreken
 - Gebruik bibliotheek netdb.h
- Functies
 - `gethostbyname`, `gethostbyaddr`
 - `getservbyname`, `getservbyport`

naam → IP-adres

- `gethostbyname()`
 - converteert een naam naar een binair IP adres
- Voorbeeld:
 - “www.amerigo.hogent.be” → 98765469
 - “192.168.16.6” → 98765469

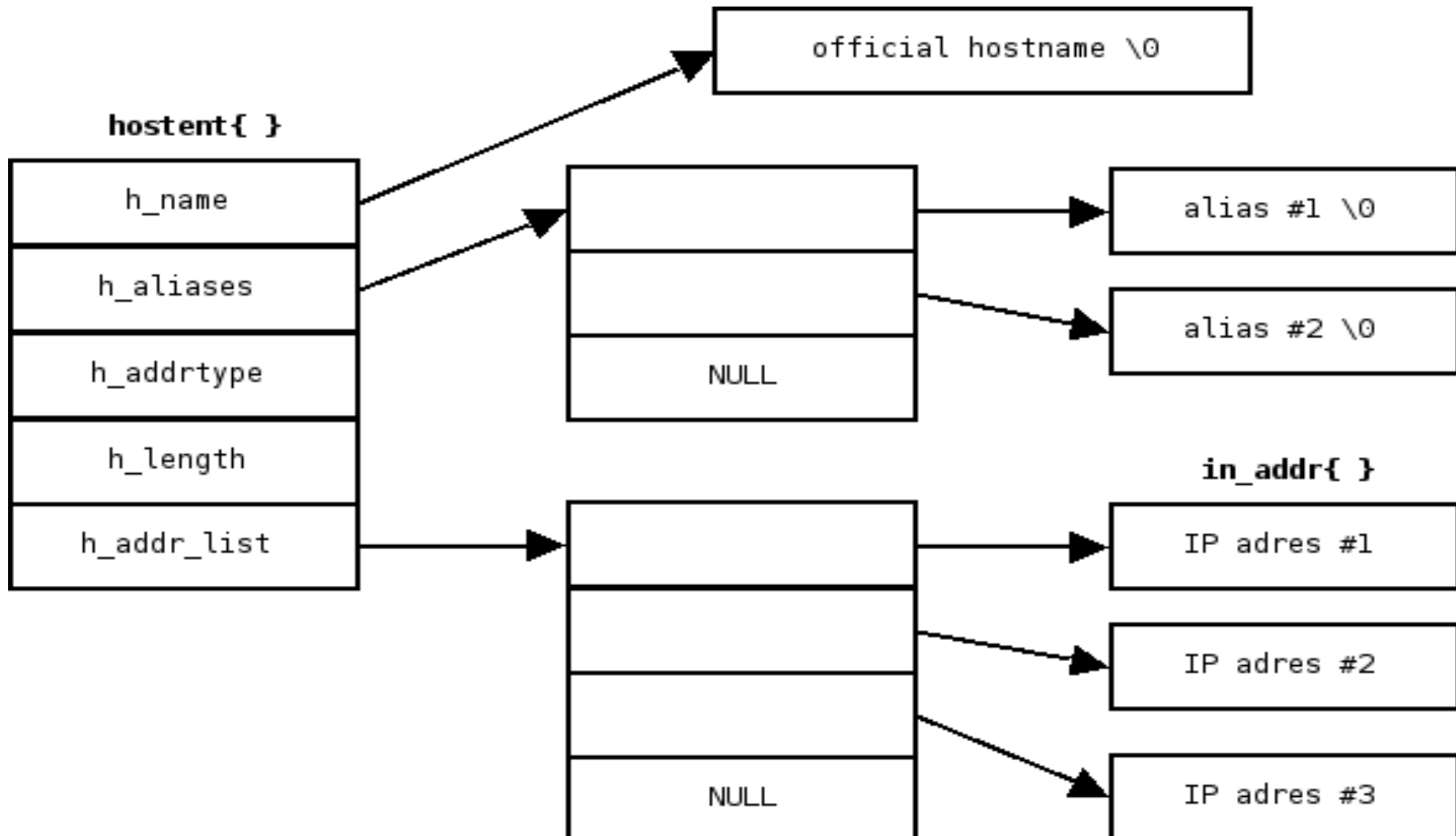
gethostbyname()

hostent ***gethostbyname**(const char *name)

```
struct hostent {  
    char    *h_name;      /* officiële naam */  
    char    **h_aliases;  /* andere namen */  
    int     h_addrtype;   /* adrestype */  
    int     h_length;     /* adres lengte */  
    char    **h_addr_list; /* IP-adressen */  
};
```

- **herror()** ipv **perror**

hostent structure



IP-adressen in hostent

- `char **h_addr_list`
 - Kan gecast naar `in_addr **`
 - IP-adressen
 - Binair
 - In netwerkvolgorde
- Indien IP-adres
 - Geen naam en aliases

Voorbeeld

- ResolveName.cpp
- Herhaling

```
struct in_addr {  
    unsigned long s_addr;  
    /* Internet address (32-bits) */  
};
```

binair en dotted decimal

- Binair IP-adres --> dotted decimal
 - `char *inet_ntoa(struct in_addr in)`
 - `char *` --> `unsigned long *`
- Herhaling
 - dotted decimal → binair IP-adres
 - `int inet_aton(char *cp, struct in_addr *inp);`

binair adres → naam

- gethostbyaddr
 - converteert een binair IP-adres naar host naam of dotted decimal
 - Voorbeeld:

98765469 → “192.168.16.6”

gethostbyaddr()

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

Return : pointer naar hostent, NULL pointer bij fout

andere functies

- `inet_addr()` : zelfde als `inet_aton()` maar verouderd omdat ze `-1` teruggeeft bij fout terwijl dit `225.225.225.225` voorstelt
- `inet_pton()` en `inet_ntop()` : zelfde als `inet_aton` en `inet_ntoa` maar ook bruikbaar voor `Ipv6`
- in de cursus komen de drie vormen voor

Opzoeken diensten

- Dienst
 - Bepaalde poort
 - `struct servent`
- Dienstnaam --> poort
 - `getservbyname`
- Poort --> Dienst
 - `getservbyport`
- Informatie uit lokale db of `/etc/services`

servent

```
struct servent {  
    char *s_name;    /* naam */  
    char **s_aliases; /* aliasesen */  
    int s_port;  
        /* poort, netwerkvolgorde */  
    char s_proto;    /*protocol*/  
}
```

getservbyname()

```
struct servent *getservbyname(  
    const char name, /*naam*/  
    const char proto /*tcp of udp*/  
);
```

Return : pointer naar servent, NULL pointer bij fout

getservbyport()

```
struct servent *getservbyport (  
    int port,  
    /* poort netwerkvolgorde */  
    const char proto /*tcp of udp*/  
);
```

Return : pointer naar servent, NULL pointer bij fout

Voorbeeld

- ResolveService.cpp
- ZoekDienst.cpp