

Network Programming:
Ch.11: Elementary Name and
Address Conversions

Li-Hsing Yen

NYCU

Ver. 1.0.1

Elementary Name and Address Conversions

- Domain name system
- `gethostbyname` Function
- `RES_USE_INET6` resolver option
- `gethostbyname2` Function and IPv6 support
- `gethostbyaddr` Function
- `uname` and `gethostname` Functions
- `getservbyname` and `getservbyport` Functions
- Other networking information

Domain Name System

- Entries in DNS: resource records (RRs) for a host
 - **A** record: maps a hostname to a 32-bit IPv4 addr
 - **AAAA** (quad A) record: maps to a 128-bit IPv6 addr
 - **PTR record**: maps IP addr to hostname (p. 11)
 - **MX record**: specifies a mail exchanger of the host
 - **CNAME** record: assigns canonical name for common services

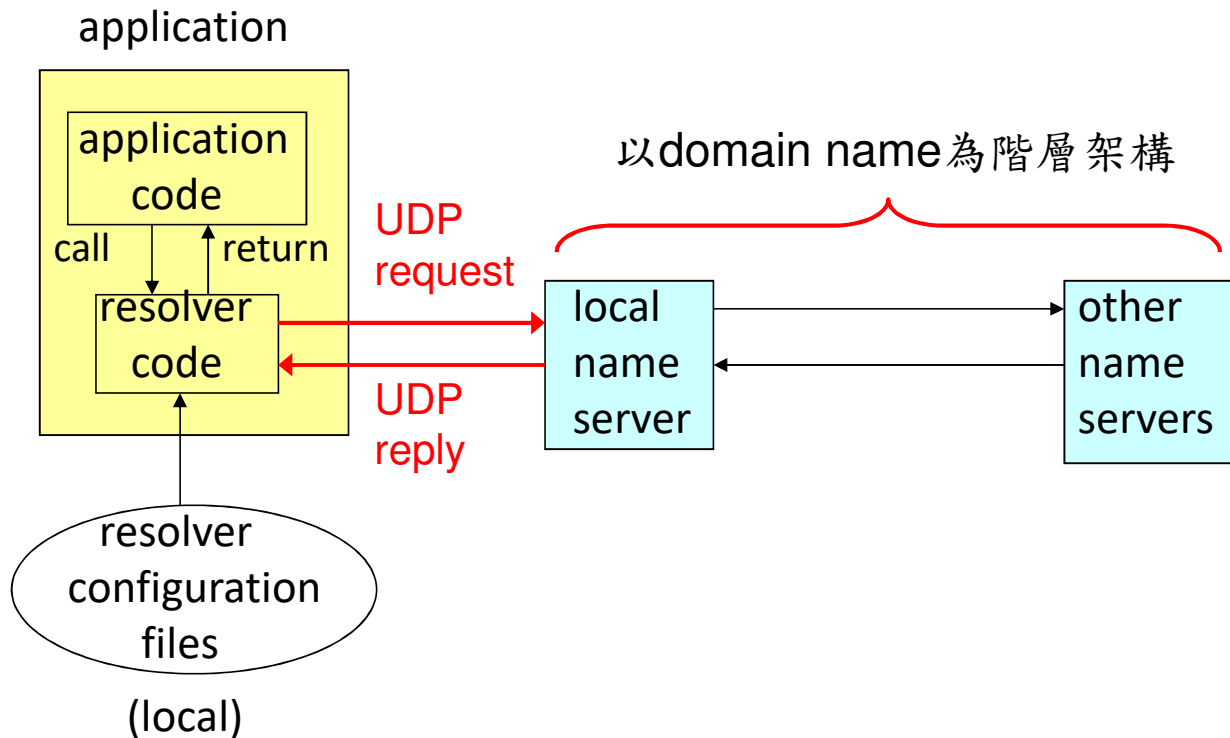
e.g.

solaris	IN	A	206.62.226.33	
	IN	AAAA	5f1b:df00:ce3e:e200:0020:0800:2078:e3e3	
	IN	MX	5 solaris.kohala.com	} 5, 10是偏好值
	IN	MX	10 mailhost.kohala.com	
www	IN	CNAME	bsdi.kohala.com	

正式名稱

5, 10是偏好值

DNS: Application, Resolver, Name Servers



resolver functions:
gethostbyname, gethostbyaddr

name server: BIND
(Berkeley Internet Name Domain)

static hosts files (DNS alternatives):
/etc/hosts

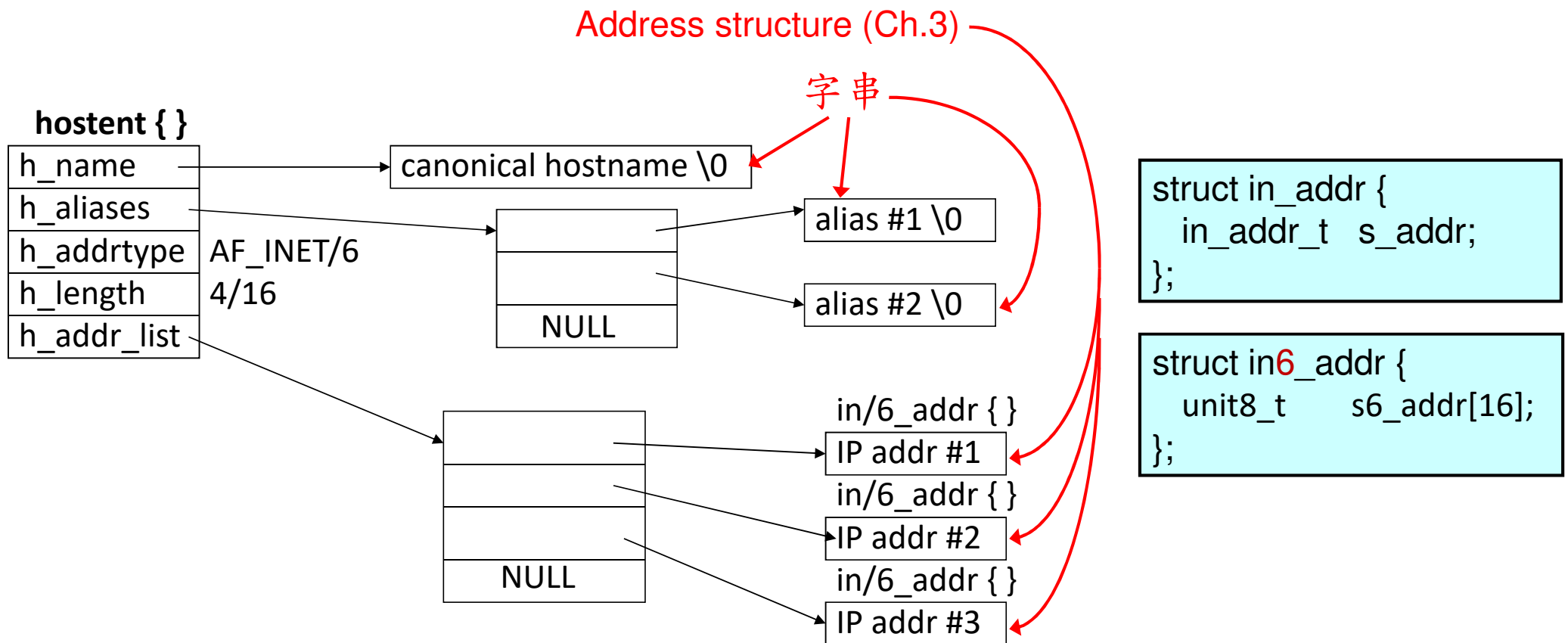
resolver configuration file
(specifies name server IPs):
/etc/resolv.conf

gethostbyname Function

performs a DNS query for an **A** record or an **AAAA** record

```
#include <netdb.h>
struct hostent *gethostbyname (const char *hostname);
    returns: a non-null pointer if OK, NULL on error with h_errno set
struct hostent {
    char *h_name;          /* official (canonical) name of host */
    char **h_aliases;     /* pointer to array of pointers to alias names */
    int  h_addrtype;      /* host addr type: AF_INET or AF_INET6 */
    int  h_length;        /* length of address: 4 or 16 */
    char **h_addr_list;   /* ptr to array of ptrs with IPv4/IPv6 addrs */
};
#define h_addr h_addr_list[0] /* first address in list */
```

hostent Structure Returned by gethostbyname



Running **hostent**

```
lhyen@lhyen-desktop:~/np/unpv13e/names$ ./hostent www.yahoo.com.tw
official hostname: src.g03.yahoodns.net
  alias: www.yahoo.com.tw
  alias: rc.yahoo.com
  address: 13.251.69.97
lhyen@lhyen-desktop:~/np/unpv13e/names$
```

Call `gethostbyname` and Print Returned Info

```
#include "unp.h" names/hostent.c

int main(int argc, char **argv)
{
    char          *ptr, **pptr, str[INET6_ADDRSTRLEN];
    struct hostent *hptr;

    while (--argc > 0) {
        ptr = *++argv;
        if ( (hptr = gethostbyname(ptr)) == NULL) {
            err_msg("gethostbyname error for host: %s: %s",
                    ptr, hstrerror(h_errno));
            continue;
        }
        printf("official hostname: %s\n", hptr->h_name);
        for (pptr = hptr->h_aliases; *pptr != NULL; pptr++)
            printf("\talias: %s\n", *pptr);
    }
}
```

下個命令列參數 → `./hostent www.nuk.edu.tw`

印出所有別名 →

印出正式名稱 →

names/hostent.c

```
switch (hptr->h_addrtype) {
case AF_INET:
#ifdef AF_INET6
case AF_INET6:
#endif
    pptr = hptr->h_addr_list;
    for ( ; *pptr != NULL; pptr++)
        printf("\taddress: %s\n",
            Inet_ntop(hptr->h_addrtype, *pptr, str, sizeof(str)));
        break;

default:
    err_ret("unknown address type");
    break;
}
}
exit(0);
}
```

呼叫 `Inet_ntop` 將所有位址全部列出來

Address type: `AF_INET` or `AF_INET6`

Reversed Name Resolution

- DNS是以domain name而不是以IP address為階層架構
- 利於由domain name查詢IP address。如果要由IP address反查domain name時不容易找到正確的DNS server
- 例如，某固網公司DNS伺服器存有所有*.idv.tw主機的domain name與IP位址的對應，但這些主機的IP位址並沒有規律性(來自各處)
- 如果給定了某個IP位址，怎能知道要去此DNS伺服器查它的domain name?

Reversed Name Resolution (cont.)

- 解決方案：建立一個特殊的domain (inaddr.arpa for IPv4)，專門用來由IP address查domain name
 - 要查206.62.226.33對應的domain name⇒到負責33.226.62.206.in-addr.arpa的DNS伺服器去查對應的PTR record
 - 處理起來就如同由domain name查IP位址

33.226.62.206.in-addr.arpa. IN PTR (admin1.uohio.edu.)

gethostbyaddr Function

binary IP address to **hostent** structure

由IP Address (置於位址結構而非字串中)查hostname (在struct hostent中)

```
#include <netdb.h>
struct hostent *gethostbyaddr (const char *addr, size_t len, int family);
returns: non-null pointer if OK, NULL on error with h_errno set
```

```
struct in_addr {
    in_addr_t s_addr;
};
```

```
struct in6_addr {
    unit8_t s6_addr[16];
};
```

- **addr** argument: a pointer to an **in_addr** or **in6_addr** structure
- **h_name** in **hostent**: canonical hostname
- **gethostbyaddr**: queries a DNS name server for a PTR record in the in-addr.arpa domain for IPv4 or a PTR record in the ip6.int domain for IPv6.

getservbyname / getservbyport Function

- Service name (如 ftp, domain, tftp) 與 port number (如21,53,69) 間的互查
- 通常定義在 /etc/services 檔中
- 同時支援TCP與UDP的服務通常在這兩個protocol內使用相同的port，但也有例外：
 - shell 514/tcp
 - syslog 514/udp

getservbyname / getservbyport Function

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

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

“ftp”, “domain”...

“udp”, “tcp” or
NULL

returns: non-null pointer if OK, NULL on error

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

returns: non-null pointer if OK, NULL on error

```
struct servent {  
    char    *s_name;        /* official service name */  
    char    **s_aliases;   /* service alias list */  
    int     s_port;        /* port number, network-byte order */  
    char    *s_proto;      /* protocol, TCP or UDP, to use */  
};
```

daytime Client

using `gethostbyname` and `getservbyname`

- (see next two pages)
- Call `gethostbyname` and `getservbyname`
- Try each server address
- Call `connect`
- Check for failure
- Read server's reply

Day Time TPC Client 2 (Specifying Server)

- argv[1] 可指定 daytime server 的 IP位址或host name

```
names/daytimetcpcli2.c
if (inet_pton(AF_INET, argv[1], &servaddr.sin_addr) == 1) {
    addrs[0] = &servaddr.sin_addr;
    addrs[1] = NULL;
    pptr = &addrs[0];
} else if ( (hp = gethostbyname(argv[1])) != NULL) {
    pptr = (struct in_addr **) hp->h_addr_list;
} else
    err_quit("hostname error for %s: %s", argv[1], hstrerror(h_errno));
```

有效IP位址

有效domain name

h_errno轉錯誤訊息

Day Time TPC Client 2 (Specifying Service)

- argv[2] 可指定 daytime server 的 port 或 service

```
if ( (n = atoi(argv[2])) > 0)
    servaddr.sin_port = htons(n);
else if ( (sp = getservbyname(argv[2], "tcp")) != NULL)
    servaddr.sin_port = sp->s_port;
else
    err_quit("getservbyname error for %s", argv[2]);
```

names/daytimetcpcli2.c

有效port num

有效service name

Other Networking Info

- Four types of info:
 - **hosts** (gethostbyname, gethostbyaddr) For IPv4
 - through DNS or /etc/hosts, hostent structure
 - **networks** (getnetbyname, getnetbyaddr)
 - through DNS or /etc/networks, netent structure
 - **protocols** (getprotobyname, getprotobynumber)
 - through /etc/protocols, protoent structure
 - **services** (getservbyname, getservbyport)
 - through /etc/services, servent structure

getaddrinfo Function

only support IPv4

Handle both name-to-address and service-to-port translation

```
#include <netdb.h>
int getaddrinfo (const char *hostname, const char *service,
                 const structure addrinfo *hints, struct addrinfo **result);
returns: 0 if OK, nonzero on error
```

hostname: either a host name or an address **string**


service: either a service name or a decimal port num **string**

hints: types of information the caller wants returned

result: a pointer to a linked list of **addrinfo** structure (next page)

addrinfo

members of
hints that can be
set by the caller



```
#include <netdb.h>

struct addrinfo {
    int      ai_flags;          /* AI_PASSIVE, AI_CANONNAME */
    int      ai_family;        /* AF_xxx */
    int      ai_socktype;      /* SOCK_xxx */
    int      ai_protocol;      /* 0 or IPPROTO_xxx for IPv4 and IPV6 */
    socklen_t ai_addrlen;
    char      *ai_canonname;
    struct sockaddr *ai_addr;
    struct addrinfo *ai_next;
};
```