

CS4254

Computer Network Architecture and Programming

Dr. Ayman A. Abdel-Hamid

Computer Science Department
Virginia Tech

Name and Address Conversions

Outline

- Name and Address Conversions (Chapter 11)
 - Domain Name System
 - gethostbyname Function
 - gethostbyaddr Function
 - gethostname Function
 - getservbyname and getservbyport Functions

Domain Name System

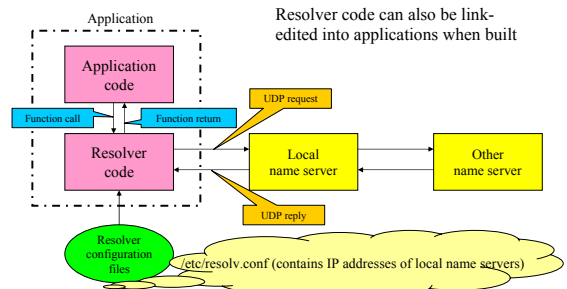
•Fully Qualified Domain Name FQDN

•DNS uses **Resource Records RR** to store information about items

•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 email
•NS	Name server	name of a server for this domain
•CNAME	Canonical name	create aliases
•PTR	Pointer	map IP addresses into host names
•HINFO	Host Description	CPU and OS in ASCII
•TXT	Text	Un-interpreted ASCII text

m.cs.vt.edu.	86400	IN	HINFO	Sun Unix
m.cs.vt.edu.	86400	IN	A	128.173.40.39
m.cs.vt.edu.	86400	IN	A	128.173.41.38

Domain Name System



Resolver code can also be link-edited into applications when built

Typical arrangement of clients, resolvers, and name servers

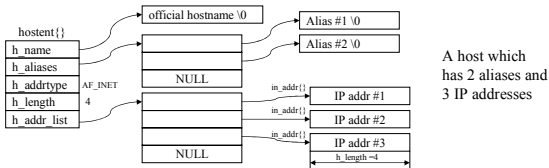
gethostbyname Function 1/2

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

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

Returns: 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 address type : AF_INET */
    int     h_length;       /* length of address : 4 */
    char    **h_addr_list; /* ptr to array of ptrs with IPv4 adrs */
};
```



Name and Address Conversions

© Dr. Ayman Abdel-Hamid, CS4254 Spring 2006

5

gethostbyname Function 2/3

```
#define h_addr h_addr_list[0] /* for backward compatibility */
```

```
struct hostent * hp = gethostbyname(argv[1]);
```

```
bcopy ( hp->h_addr, &server.sin_addr, hp->h_length);
```

```
//see intro/daytimetcpcli_hostname.c
```

- Will only retrieve IPv4 addresses, performs a query for an A record

- Some versions of `gethostbyname` will allow the following

```
hptr = gethostbyname ("192.168.42.2"); → not portable
```

Name and Address Conversions

© Dr. Ayman Abdel-Hamid, CS4254 Spring 2006

6

gethostbyname Function 3/3

- If error, sets global integer `h_errno` to

- `HOST_NOT_FOUND`

- `TRY_AGAIN`

- `NO_RECOVERY`

- `NO_DATA` → specified name valid but does not have A records

- Can use `hstrerror` function to get a description of the error (value of `h_errno`)

- See `names/hostent.c` for an example

- Example Usage

```
>hostent ap1      >hostent cnn.com    >hostent www
```

Name and Address Conversions

© Dr. Ayman Abdel-Hamid, CS4254 Spring 2006

7

gethostbyaddr Function

- Takes a binary IPv4 address and tries to find the hostname corresponding to that address

- Performs a query for a PTR record

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

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

Returns non-null pointer if OK, NULL on error with `h_errno` set

- Field of interest in the returning structure is `h_name` (canonical host name)

- `addr` argument is not a `char*` but really a pointer to an `in_addr` structure containing the IPv4 address

Name and Address Conversions

© Dr. Ayman Abdel-Hamid, CS4254 Spring 2006

8

gethostname Function

•Obtains the host name

```
#include <unistd.h>
```

```
int gethostname(char *name, size_t len);
```

// On success, zero is returned. On error, -1 is returned, and *errno* is set appropriately

•Example

```
#define MAXHOSTNAME 80
```

```
char ThisHost[80];
```

```
gethostname (ThisHost, MAXHOSTNAME);
```

getservbyname and getservbyport Functions ^{1/2}

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

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

//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

//port value must be in network byte order

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

getservbyname and getservbyport Functions ^{2/2}

```
struct servent *sptr;
```

```
sptr = getservbyname ("domain", "udp"); //DNS using UDP
```

```
sptr = getservbyname ("ftp", "tcp"); //FTP using TCP
```

```
sptr=getservbyname("ftp", "udp"); //this call will fail
```

```
sptr = getsrvbyport(htons(21), "tcp"); // FTP using TCP
```

```
sptr = getsrvbyport(htons(21), NULL); // FTP using TCP
```

```
sptr = getsrvbyport(htons(21), "udp"); // This call will fail
```

•See `names/daytimetcpcli1.c` for a program that takes a hostname and service name as arguments