DNS NOTES

**arpa/inet.h**

inet\_ntoa

convert numeric IP their binary <-> text forms.

inet\_ntoa

IPv4 binary -> string

in the standard dot notation (192.168.1.1).

**netdb.h**

define the structures, macros, functions

used for domain name resolution.

include definition for gethostbyname, gethostbyaddr, hostent.

gethostbyname

retrieve info about a host based on its domain name.

It returns a pointer to a hostent structure

that contains info about the host such as its IP addresses.

A black background with white text

Description automatically generated

perform a DNS lookup for a given domain name,

converting it into one or more IP addresses

struct hostent iceririk

A screen shot of a computer

Description automatically generated

char \*h\_name; host name.

char \*\*h\_aliases; alternate names array.

int h\_addrtype; Address type (AF\_INET for IPv4).

int h\_length; address Length.

char \*\*h\_addr\_list; network addresses array.

**\*host\_entry**

pointer to a struct hostent

used to store info about a given host (such as a domain name or IP address).

**gethostbyname()**

When the function **gethostbyname()** is called with a domain name,

it performs a DNS lookup and returns a pointer to a struct hostent

that contains info about the host.

This info includes

the host's name, aliases, address type, length of the address,

a list of addresses associated with the host.

**struct hostent \*host\_entry;**

declare a pointer to a struct hostent.

struct hostent is a structure defined in the <netdb.h> header,

which stores info about a given host (such as its IP addresses).

The host\_entry pointer

will later point to the data returned by the gethostbyname function.

**host\_entry = gethostbyname( domain.c\_str() );**

gethostbyname funct

perform DNS lookup.

It takes a domain name as an argument

and returns a pointer to a struct hostent containing information about the host.

domain.c\_str()

domain string -> C-style string (const char\*)

gethostbyname expects a C-style string.

A computer code on a black background

Description automatically generated

**Initialization: char \*ip\_address;**

declares a pointer to a character array

that will store the IP address in a human-readable format.

h\_addr\_list[i]

part of the hostent structure,

store host info (such as a domain name) obtained via DNS lookup.

struct hostent

{

char \*\*h\_addr\_list; // Null-terminated network addresses array for the host.

};

**Loop through IP addresses: for (int i = 0;** host\_entry -> h\_addr\_list[i] != NULL; **i++)**

This for loop iterates through the list of IP addresses associated with the domain name.

char \*\*h\_addr\_list

vector of addresses for the host.

host might be connected to multiple networks and have different addresses on each one.

host\_entry -> h\_addr\_list

array of pointers to network addresses for the given host (domain).

host\_entry->h\_addr\_list[i] != NULL

checks that the current address pointer is not null,

ensuring that the loop continues until all addresses have been processed.

A black background with white text

Description automatically generated

**Convert network address to readable format:**

**ip\_address = inet\_ntoa( \* (struct in\_addr \*) host\_entry -> h\_addr\_list[i] );**

host\_entry -> h\_addr\_list[i]

gives a pointer to a network address.

(struct in\_addr \*) host\_entry -> h\_addr\_list[i]

casts this pointer to a pointer of type struct in\_addr.

inet\_ntoa expects an argument of type struct in\_addr.

\* dereferences the pointer,

giving the actual struct in\_addr object.

inet\_ntoa converts this struct in\_addr object

to a string in dotted-decimal format (e.g., "192.168.1.1").

----

**in\_addr**

in\_addr structure

represents an IPv4 address in the binary format.

defined in the <netinet/in.h> header file.

structure contains a single member,

which is an unsigned 32-bit integer that holds the IPv4 address.

Here is the typical definition of the in\_addr structure:

A black screen with white text

Description automatically generated

A computer screen with white text

Description automatically generated

In the code you provided,

the inet\_ntoa function is used to convert an in\_addr structure

to a human-readable string representing the IPv4 address in the dotted-decimal format.

----

**->** gecerli olmasi icin

Degisken class - structure icinde veya

Pointer olmalı



A screen shot of a computer program

Description automatically generated