NAME

gethostbyname, **gethostbyname2**, **gethostbyaddr**, **gethostent**, **sethostent**, **endhostent**, **herror**, **hstrerror**, **gethostbyname_r**, **gethostbyname2_r**, **gethostbyaddr_r** - get network host entry

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <netdb.h>

int h_errno;

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

*struct hostent **
gethostbyname2(*const char *name, int af*);

*struct hostent **
gethostbyaddr(*const void *addr, socklen_t len, int af*);

*struct hostent **
gethostent(*void*);

void
sethostent(*int stayopen*);

void
endhostent(*void*);

void
herror(*const char *string*);

*const char **
hstrerror(*int err*);

int
gethostbyname_r(*const char *name, struct hostent *he, char *buffer, size_t buflen,*
*struct hostent **result, int *h_errnop*);

int
gethostbyname2_r(*const char *name, int af, struct hostent *he, char *buffer, size_t buflen, struct hostent **result, int *h_errnop*);

int
gethostbyaddr_r(*const void *addr, socklen_t len, int af, struct hostent *hp, char *buf, size_t buflen, struct hostent **result, int *h_errno, p*);

DESCRIPTION

The getaddrinfo(3) and getnameinfo(3) functions are preferred over the gethostbyname(), gethostbyname2(), and gethostbyaddr() functions.

The **gethostbyname**(), **gethostbyname2**() and **gethostbyaddr**() functions each return a pointer to an object with the following structure describing an internet host referenced by name or by address, respectively.

The *name* argument passed to **gethostbyname**() or **gethostbyname2**() should point to a NUL-terminated hostname. The *addr* argument passed to **gethostbyaddr**() should point to an address which is *len* bytes long, in binary form (i.e., not an IP address in human readable ASCII form). The *af* argument specifies the address family (e.g. AF_INET, AF_INET6, etc.) of this address.

The structure returned contains either the information obtained from the name server, broken-out fields from a line in */etc/hosts*, or database entries supplied by the yp(8) system. The order of the lookups is controlled by the ‘hosts’ entry in nsswitch.conf(5).

```
struct    hostent {
          char    *h_name;/* official name of host */
          char    **h_aliases;      /* alias list */
          int     h_addrtype;       /* host address type */
          int     h_length; /* length of address */
          char    **h_addr_list;    /* list of addresses from name server */
};
#define   h_addr  h_addr_list[0]    /* address, for backward compatibility */
```

The members of this structure are:

h_name Official name of the host.

h_aliases A NULL-terminated array of alternate names for the host.

h_addrtype The type of address being returned; usually AF_INET.

h_length The length, in bytes, of the address.

h_addr_list A NULL-terminated array of network addresses for the host. Host addresses are returned in network byte order.

h_addr The first address in *h_addr_list*; this is for backward compatibility.

When using the nameserver, **gethostbyname**() and **gethostbyname2**() will search for the named host in the current domain and its parents unless the name ends in a dot. If the name contains no dot, and if the environment variable "HOSTALIASES" contains the name of an alias file, the alias file will first be searched for an alias matching the input name. See hostname(7) for the domain search procedure and the alias file format.

The **gethostbyname2**() function is an evolution of **gethostbyname**() which is intended to allow lookups in address families other than AF_INET, for example AF_INET6.

The **sethostent**() function may be used to request the use of a connected TCP socket for queries. Queries will by default use UDP datagrams. If the *stayopen* flag is non-zero, a TCP connection to the name server will be used. It will remain open after calls to **gethostbyname**(), **gethostbyname2**() or **gethostbyaddr**() have completed.

The **endhostent**() function closes the TCP connection.

The **herror**() function writes a message to the diagnostic output consisting of the string argument *string*, the constant string ": ", and a message corresponding to the value of *h_errno*.

The **hstrerror**() function returns a string which is the message text corresponding to the value of the *err* argument.

Functions with the *_r* suffix provide reentrant versions of their respective counterparts. The caller must supply five additional parameters: a *struct hostent* variable to be filled on success, a *buffer* of *buflen* bytes in size, a *struct hostent result* variable that will point to the result on success or be set to NULL on failure or if the name is not found. The *h_errnop* variable will be filled with the error code if any. All these functions return 0 on success.

FILES

/etc/hosts
/etc/nsswitch.conf
/etc/resolv.conf

EXAMPLES

Print out the hostname associated with a specific IP address:

```
const char *ipstr = "127.0.0.1";
struct in_addr ip;
struct hostent *hp;

if (!inet_aton(ipstr, &ip))
        errx(1, "can't parse IP address %s", ipstr);

if ((hp = gethostbyaddr((const void *)&ip,
   sizeof ip, AF_INET)) == NULL)
        errx(1, "no name associated with %s", ipstr);

printf("name associated with %s is %s\n", ipstr, hp->h_name);
```

DIAGNOSTICS

Error return status from **gethostbyname**(), **gethostbyname2**() and **gethostbyaddr**() is indicated by return of a NULL pointer. The integer *h_errno* may then be checked to see whether this is a temporary failure or an invalid or unknown host. The routine **herror**() can be used to print an error message describing the failure. If its argument *string* is non-NULL, it is printed, followed by a colon and a space. The error message is printed with a trailing newline.

The variable *h_errno* can have the following values:

HOST_NOT_FOUND No such host is known.

TRY_AGAIN This is usually a temporary error and means that the local server did not receive a response from an authoritative server. A retry at some later time may succeed.

NO_RECOVERY Some unexpected server failure was encountered. This is a non-recoverable error.

NO_DATA The requested name is valid but does not have an IP address; this is not a temporary error. This means that the name is known to the name server but there is no address associated with this name. Another type of request to the name server using this domain name will result in an answer; for example, a mail-forwarder may be registered for this domain.

SEE ALSO

getaddrinfo(3), getnameinfo(3), inet_aton(3), resolver(3), hosts(5), hostname(7)

HISTORY

The **herror**() function appeared in 4.3BSD. The **endhostent**(), **gethostbyaddr**(), **gethostbyname**(), **gethostent**(), and **sethostent**() functions appeared in 4.2BSD. The **gethostbyname2**() function first appeared in BIND version 4.9.4. The **gethostbyname_r**() function first appeared in FreeBSD 6.2.

CAVEATS

The **gethostent**() function is defined, and **sethostent**() and **endhostent**() are redefined, when Standard C Library (libc, -lc) is built to use only the routines to lookup in */etc/hosts* and not the name server.

The **gethostent**() function reads the next line of */etc/hosts*, opening the file if necessary.

The **sethostent**() function opens and/or rewinds the file */etc/hosts*. If the *stayopen* argument is non-zero, the file will not be closed after each call to **gethostbyname**(), **gethostbyname2**() or **gethostbyaddr**().

The **endhostent**() function closes the file.

BUGS

These functions use a thread-specific data storage; if the data is needed for future use, it should be copied before any subsequent calls overwrite it.

Though these functions are thread-safe, still it is recommended to use the getaddrinfo(3) family of functions, instead.

Only the Internet address format is currently understood.