

**NOM**

gethostbyname, gethostbyaddr, sethostent, gethostent, endhostent, herror, hstrerror – Obtenir des informations concernant le réseau.

**SYNOPSIS**

```
#include <netdb.h>
extern int h_errno;

struct hostent *gethostbyname(const char *name);

#include <sys/socket.h> /* pour avoir AF_INET */
struct hostent *
gethostbyaddr(const void *addr, int len, int type);

void sethostent(int stayopen);

void endhostent(void);

void herror(const char *s);

const char * hstrerror(int err);

/* extension Système V/POSIX */
struct hostent *gethostent(void);

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

int gethostent_r(
    struct hostent *ret, char *buf, size_t buflen,
    struct hostent **result, int *h_errnop);

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

int gethostbyname2_r(const char *name, int af,
    struct hostent *ret, char *buf, size_t buflen,
    struct hostent **result, int *h_errnop);
```

**DESCRIPTION**

La fonction **gethostbyname()** renvoie une structure de type *hostent* pour l'hôte *name*. La chaîne *name* est soit un nom d'hôte, soit une adresse IPv4 en notation pointée standard, soit une adresse IPv6 avec la notation points-virgules et points (Cf RFC 1884 pour la description des adresses IPv6). Si *name* est une adresse IPv4 ou IPv6, aucune recherche supplémentaire n'a lieu et **gethostbyname()** copie simplement la chaîne *name* dans le champ *h\_name* et le champ équivalent *struct in\_addr* dans le champ *h\_addr\_list[0]* de la structure *hostent* renvoyée.

Si *name* ne se termine pas par un point, et si la variable d'environnement **HOSTALIASES** est configurée, le fichier d'alias indiqué par **HOSTALIASES** sera d'abord parcouru à la recherche de *name* (voir **hostname(7)** pour le format du fichier). Le domaine courant et ses parents sont parcourus si *name* ne se termine pas par un point.

La fonction **gethostbyaddr()** renvoie une structure du type *hostent* pour l'hôte d'adresse *addr*. Cette

adresse est de longueur *len* et du *type* donné. Les types d'adresse valides sont **AF\_INET** et **AF\_INET6**. L'argument adresse de l'hôte est un pointeur vers une structure de type dépendant du type de l'adresse, par exemple **struct in\_addr \*** (probablement obtenu via un appel à **inet\_addr()**) pour une adresse de type **AF\_INET**.

La fonction **sethostent ()** indique, si *stayopen* est vrai (vaut 1), qu'une socket TCP connectée doit être utilisée pour interroger le serveur de noms et que la connexion doit rester ouverte durant les demandes successives. Sinon l'interrogation utilisera des datagrammes UDP.

La fonction **endhostent()** ferme la socket TCP connectée utilisée pour interroger le serveur de noms du domaine.

La fonction (obsolète) **herror()** affiche le message d'erreur associé avec la valeur courante de *h\_errno* sur la sortie standard `stderr`.

La fonction (obsolète) **herror()** reçoit un numéro d'erreur en argument (typiquement *h\_errno*) et renvoie la chaîne de message d'erreur.

Les interrogations du serveur de noms effectuées par **gethostbyname()** et **gethostbyaddr()** utilisent les éléments suivants : le serveur de noms **named(8)**, les lignes de */etc/hosts*, et l'annuaire Network Information Service (NIS ou YP), suivant le contenu de la ligne *order* du fichier */etc/host.conf*. L'action par défaut consiste à interroger **named(8)**, puis */etc/hosts*.

La structure *hostent* est définie ainsi dans *<netdb.h>* :

```
struct hostent {
    char *h_name; /* Nom officiel de l'hôte. */
    char **h_aliases; /* Liste d'alias. */
    int h_addrtype; /* Type d'adresse de l'hôte. */
    int h_length; /* Longueur de l'adresse. */
    char **h_addr_list; /* Liste d'adresses. */
}
#define h_addr h_addr_list[0] /* pour retro-compatibilité. */
```

Les membres de la structure *hostent* sont :

*h\_name*

Nom officiel de l'hôte.

*h\_aliases*

Une table d'alternatives au nom officiel de l'hôte, terminée par un pointeur NULL.

*h\_addrtype*

Le type d'adresse : toujours **AF\_INET** ou **AF\_INET6**.

*h\_length*

La longueur, en octets, de l'adresse.

*h\_addr\_list*

Une table, terminée par un pointeur NULL, d'adresses réseau pour l'hôte, avec l'ordre des octets du réseau.

*h\_addr* La première adresse dans *h\_addr\_list* pour respecter la compatibilité ascendante.

## VALEUR RENVOYÉE

Les fonctions **gethostbyname()** et **gethostbyaddr()** renvoient un pointeur sur la structure *hostent*, ou bien un pointeur NULL si une erreur se produit, auquel cas *h\_errno* contient le code d'erreur. Lorsqu'elle n'est pas NULL, la valeur de retour peut pointer sur une donnée statique. Voir les notes plus loin.

## ERREURS

La variable *h\_errno* peut prendre les valeurs suivantes :

**HOST\_NOT\_FOUND**

L'hôte indiqué est inconnu.

**NO\_ADDRESS** ou **NO\_DATA**

Le nom est valide mais ne possède pas d'adresse IP.

**NO\_RECOVERY**

Une erreur fatale du serveur de noms est apparue.

**TRY\_AGAIN**

Une erreur temporaire du serveur de noms est apparue, essayez un peu plus tard.

**FICHIERS**

*/etc/host.conf*

Fichier de configuration de la résolution de noms.

*/etc/hosts*

Base de données des hôtes.

*/etc/nsswitch.conf*

Configuration du commutateur de service de noms.

**CONFORMITÉ**

BSD 4.3, POSIX.1-2001.

**EXTENSION SYSTÈME V/POSIX**

POSIX réclame l'appel **gethostent()**, qui devrait renvoyer la nouvelle entrée de la base de données des hôtes. Lorsqu'on utilise DNS/BIND, cela n'a pas de sens, mais il peut être raisonnable si la base de données des hôtes est un fichier qui peut être lu ligne par ligne. Sur beaucoup de systèmes, une routine du même nom lit le fichier */etc/hosts*. Il peut être disponible seulement si la bibliothèque a été compilée sans le support DNS. La version glibc ignore les entrées ipv6. Cette fonction n'est pas réentrante mais la glibc en possède une : **gethostent\_r()**.

**EXTENSIONS GNU**

La glibc2 propose aussi une fonction **gethostbyname2()** qui agit comme **gethostbyname()**, qui permet de préciser la famille à laquelle l'adresse doit appartenir.

La glibc2 propose aussi les versions réentrantes **gethostbyname\_r()** et **gethostbyname2\_r()**. Elles renvoient zéro si elles réussissent et une valeur non-nulle en cas d'erreur. Le résultat de l'appel est stocké dans la structure d'adresse *ret*. Après l'appel, *\*result* vaudra NULL en cas d'erreur, ou pointera sur le résultat. Des données auxiliaires seront stockées dans le tampon *buf* de longueur *buflen*. (Si le tampon est trop petit, ces fonctions renverront **ERANGE**). La variable *h\_errno* n'est pas modifiée, mais l'adresse d'une variable où stocker le code d'erreur est transmis dans *h\_errnop*.

**NOTES**

Les fonctions **gethostbyname()** et **gethostbyaddr()** peuvent renvoyer des pointeurs sur des données statiques susceptibles d'être écrasées d'un appel à l'autre. Copier la structure *struct hostent* ne suffit pas car elle contient elle-même des pointeurs. Une copie en profondeur est indispensable.

Les spécifications SUS-v2 déclarent – à tort – le paramètre *len* de **gethostbyaddr()** de type *size\_t*. (Ceci est erroné car il doit obligatoirement être un *int*, ce que *size\_t* n'est pas toujours. POSIX.1-2001 le déclare *socklen\_t*, ce qui est correct).

L prototype BSD pour **gethostbyaddr()** utilise *const char \** comme premier argument.

POSIX.1-2001 indique **gethostbyaddr()** et **gethostbyname()** comme obsolètes. Voir **getaddrinfo(3)**, **getnameinfo(3)**, **gai\_strerror(3)**.

**VOIR AUSSI**

**getaddrinfo(3)**, **getipnodebyaddr(3)**, **getipnodebyaddr(3)**, **getnameinfo(3)**, **inet\_ntop(3)**, **inet\_pton(3)**, **resolver(3)**, **hosts(5)**, **nsswitch.conf(5)**, **hostname(7)**, **named(8)**

**TRADUCTION**

Ce document est une traduction réalisée par Christophe Blaess <<http://www.blaess.fr/christophe/>> le 26 octobre 1996 et révisée le 14 août 2006.

L'équipe de traduction a fait le maximum pour réaliser une adaptation française de qualité. La version anglaise la plus à jour de ce document est toujours consultable via la commande : « **LANG=C man 3 gethostbyname** ». N'hésitez pas à signaler à l'auteur ou au traducteur, selon le cas, toute erreur dans cette page de manuel.