NOM

connect − Débuter une connexion sur une socket.

SYNOPSIS

#include <sys/types.h>
#include <sys/socket.h>

int connect(int *sockfd***, struct sockaddr ****serv_addr***, socklen_t** *addrlen***);**

DESCRIPTION

L'appel système **connect**() connecte la socket référencée par le descripteur de fichier *sockfd* à l'adresse spécifiée par *serv_addr*. L'argument *addrlen* spécifie la taille de *serv_addr*. Le format de l'adresse dans *serv_addr* est déterminé par l'espace adresse de la socket *sockfd* ; voir **socket**(2) pour plus de détails. Si la socket est du type **SOCK_DGRAM**, alors *serv_adr* est l'adresse à laquelle les datagrammes seront envoyés par défaut, et la seule adresse depuis laquelle ils seront reçus. Si la socket est du type **SOCK_STREAM** ou **SOCK_SEQPACKET**, cette fonction tente de se connecter à la socket qui est liée à l'adresse indiquée par *serv_addr*.

En général, les sockets des protocoles orientés connexion ne réussissent un appel **connect**() qu'une seule fois, alors qu'une socket d'un protocole sans connexion peut appeler **connect**() plusieurs fois pour changer son affectation. Une socket sans connexion peut interrompre son affectation en se connectant sur une adresse avec le membre *sa_family* de la structure **sockaddr** à la valeur **AF_UNSPEC**.

VALEUR RENVOYÉE

connect() renvoie 0 s'il réussit, ou −1 s'il échoue, auquel cas *errno* contient le code d'erreur.

ERREURS

Voici une liste d'erreurs générales concernant les sockets, il peut en exister d'autres spécifiques au domaine employé.

EACCES
Pour les sockets de domaine Unix qui sont spécifiées par un nom de chemin : la permission en écriture est refusée sur le fichier socket, ou la permission de parcours est refusée pour l'un des répertoires composant le chemin. (Voir aussi **path_resolution**(2).)

EACCES, **EPERM**
L'utilisateur a tenté de connecter une adresse broadcast sans avoir activé l'attribut broadcast, ou la demande de connexion a échoué à cause des règles d'un firewall local.

EADDRINUSE
L'adresse est déjà utilisée.

EAFNOSUPPORT
L'adresse transmise n'appartient pas à la famille indiquée dans son champ *sa_family*.

EAGAIN
Pas de port local disponible, ou pas assez de place dans les tables de routage. Pour **PF_INET** voir l'appel sysctl **net.ipv4.ip_local_port_range** dans **ip**(7) pour savoir comment augmenter le nombre de ports locaux.

EALREADY
La socket est non bloquante et une tentative de connexion précédente ne s'est pas encore terminée.

EBADF
Mauvais descripteur.

ECONNREFUSED
La connexion est refusée par le serveur.

EFAULT
La structure d'adresse pointe en dehors de l'espace d'adressage.

EINPROGRESS
La socket est non bloquante, et la connexion ne peut pas être établie immédiatement. Il est alors possible d'utiliser **select**(2) ou **poll**(2) pour attendre que la socket soit disponible en écriture. Une fois que **select**() confirme la possibilité d'écrire, utilisez **getsockopt**(2) pour lire l'option **SO_ERROR** du niveau **SOL_SOCKET** et déterminer si **connect**() s'est terminé avec succès (**SO_ERROR** vaut zéro) ou en échec (**SO_ERROR** contient l'un des codes d'erreurs listés ici, indiquant le problème).

EINTR
L'appel système a été interrompu par un signal qui a été intercepté.

EISCONN
La socket est déjà connectée.

ENETUNREACH
Le réseau est inaccessible.

ENOTSOCK
Le descripteur ne correspond pas à une socket.

ETIMEDOUT
Dépassement du délai maximum pendant la connexion. Le serveur peut être trop chargé pour accepter une nouvelle connexion. Remarquez que pour les sockets IP, le délai peut être très long si les syncookies sont activés sur le serveur.

CONFORMITÉ
SVr4, BSD 4.4 (La fonction **connect**() est apparue en premier dans BSD 4.2).

NOTE
Le troisième argument de **connect**() est en fait un *int*
et c'est ce qu'utilisent BSD 4.*, libc4 et libc5). Une certaine confusion POSIX résulte du "socklen_t" actuel. Les propositions de standard n'ont pas encore été adoptées, mais glibc2 les suit déjà et utilise *socklen_t*. Pour plus de détails voir **accept**(2).

BOGUES
La déconnexion d'une socket en appelant **connect**() avec un adresse de type **AF_UNSPEC** n'est pas encore implémentée.

VOIR AUSSI
accept(2), **bind**(2), **getsockname**(2), **listen**(2), **path_resolution**(2), **socket**(2)

TRADUCTION
Ce document est une traduction réalisée par Christophe Blaess <http://www.blaess.fr/christophe/> le 10 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 2 connect** ». N'hésitez pas à signaler à l'auteur ou au traducteur, selon le cas, toute erreur dans cette page de manuel.