

NOM

`bind` – Fournir un nom à une socket.

SYNOPSIS

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

```
int bind(int sockfd, const struct sockaddr *my_addr, socklen_t addrlen);
```

DESCRIPTION

`bind()` fournit à la socket `sockfd`, l'adresse locale `my_addr`. `my_addr` est longue de `addrlen` octets. Traditionnellement cette opération est appelée « affectation d'un nom à une socket » (Quand une socket est créée, par l'appelsystème `socket(2)`, elle existe dans l'espace des noms mais n'a pas de nom assigné).

Il est normalement nécessaire d'affecter une adresse locale avec `bind()` avant qu'une socket `SOCK_STREAM` puisse recevoir des connexions (voir `accept(2)`).

Les règles d'affectation de nom varient suivant le domaine de communication. Consultez le manuel Linux section 7 pour de plus amples informations. Pour `AF_INET` voir `ip(7)`, pour `AF_INET6` voir `ipv6(7)`, pour `AF_UNIX` voir `unix(7)`, pour `AF_APPLETALK` voir `ddp(7)`, pour `AF_PACKET` voir `packet(7)`, pour `AF_X25` voir `x25(7)` et pour `AF_NETLINK` voir `netlink(7)`. La structure actuelle passée comme argument `my_addr` dépendra de la famille d'adresses. La définition de la structure `sockaddr` est quelque chose comme :

```
struct sockaddr {
    sa_family_t sa_family;
    char      sa_data[14];
}
```

Le seul objectif de cette structure est de transtyper le pointeur structure passé dans `my_addr` afin d'éviter les avertissements à la compilation. L'exemple suivant montre comment c'est fait lorsqu'on attache une socket dans le domaine Unix (`AF_UNIX`) :

```
#include <sys/socket.h>
#include <sys/un.h>
#include <stdlib.h>
#include <stdio.h>

#define MY_SOCKET_PATH "/unchemin"

int
main(int argc, char *argv[])
{
    int sfd;
    struct sockaddr_un addr;

    sfd = socket(AF_UNIX, SOCK_STREAM, 0);
    if (sfd == -1) {
        perror("socket");
        exit(EXIT_FAILURE);
    }

    memset(&addr, 0, sizeof(struct sockaddr_un));
        /* Clear structure */
    addr.sun_family = AF_UNIX;
    strncpy(addr.sun_path, MY_SOCKET_PATH,
```

```

        sizeof(addr.sun_path) - 1);

    if (bind(sfd, (struct sockaddr *) &addr,
        sizeof(struct sockaddr_un)) == -1) {
        perror("bind");
        exit(EXIT_FAILURE);
    }
    ...
}

```

VALEUR RENVOYÉE

L'appel renvoie 0 s'il réussit, ou -1 s'il échoue, auquel cas *errno* contient le code d'erreur.

ERREURS

EACCES

L'adresse est protégée et l'utilisateur n'est pas le super-utilisateur.

EADDRINUSE

L'adresse fournie est déjà utilisée.

EBADF

sockfd n'est pas un descripteur valide.

EINVAL

La socket possède déjà une adresse. pour les détails.

ENOTSOCK

sockfd est un descripteur de fichier, pas une socket.

Les erreurs suivantes sont spécifiques aux sockets du domaine UNIX (**AF_UNIX**) :

EACCES

L'accès à un composant du chemin d'accès n'est pas autorisé. (Voir aussi **path_resolution(2)**.)

EADDRNOTAVAIL

Une interface non existante a été demandée ou l'adresse demandée n'était pas locale.

EFAULT

my_addr pointe en dehors de l'espace d'adresse accessible.

EINVAL

La longueur *addr_len* est fautive, ou la socket n'est pas de la famille **AF_UNIX**.

ELOOP

my_addr contient des références circulaires (à travers un lien symbolique).

ENAMETOOLONG

my_addr est trop long

ENOENT

Le fichier n'existe pas.

ENOMEM

pas assez de mémoire pour le noyau.

ENOTDIR

Un composant du chemin d'accès n'est pas un répertoire.

EROFS

L'i-noeud se trouverait dans un système de fichiers en lecture seule.

BOGUES

Les options de proxy transparent ne sont pas décrites.

CONFORMITÉ

SVr4, BSD 4.4 (l'appel système **bind()** est apparu dans BSD 4.2).

NOTE

Le troisième argument de **bind()** 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, également utilisé dans la glibc. Voir **accept(2)**.

VOIR AUSSI

accept(2), **connect(2)**, **getsockname(2)**, **listen(2)**, **path_resolution(2)**, **socket(2)**, **getaddrinfo(3)**, **ip(7)**, **ipv6(7)**, **socket(7)**, **unix(7)**

TRADUCTION

Ce document est une traduction réalisée par Christophe Blaess <<http://www.blaess.fr/christophe/>> le 9 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 bind** ». N'hésitez pas à signaler à l'auteur ou au traducteur, selon le cas, toute erreur dans cette page de manuel.