Prev Contents Next

bind()

Associate a socket with an IP address and port number

Prototypes

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

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

Description

When a remote machine wants to connect to your server program, it needs two pieces of information: the IP address and the port number. The `bind()` call allows you to do just that.

First, you call `socket()` to get a socket descriptor, and then you load up a `struct sockaddr_in` with the IP address and port number information, and then you pass both of those into `bind()`, and the IP address and port are magically (using actual magic) bound to the socket!

If you don't know your IP address, or you know you only have one IP address on the machine, or you don't care which of the machine's IP addresses is used, you can simply set the `s_addr` field in your `struct sockaddr_in` to **`INADDR_ANY`** and it will fill in the IP address for you.

Lastly, the **`addrlen`** parameter should be set to `sizeof(my_addr)`.

Return Value

Returns zero on success, or **`-1`** on error (and `errno` will be set accordingly.)

Example

```
struct sockaddr_in myaddr;
int s;

myaddr.sin_family = AF_INET;
myaddr.sin_port = htons(3490);

// you can specify an IP address:
inet_aton("63.161.169.137", &myaddr.sin_addr.s_addr);

// or you can let it automatically select one:
myaddr.sin_addr.s_addr = INADDR_ANY;

s = socket(PF_INET, SOCK_STREAM, 0);
bind(s, (struct sockaddr*)myaddr, sizeof(myaddr));
```

See Also

`socket()`, `struct sockaddr_in`, `struct in_addr`

Prev Contents Next