

NAME

bindresvport - bind a socket to a privileged IP port

SYNOPSIS

```
#include <sys/types.h>
#include <netinet/in.h>
```

```
int bindresvport(int sockfd, struct sockaddr_in *sin);
```

DESCRIPTION

bindresvport() is used to bind a socket descriptor to a privileged anonymous IP port, that is, a port number arbitrarily selected from the range 512 to 1023.

If the **bind(2)** performed by **bindresvport()** is successful, and *sin* is not NULL, then *sin->sin_port* returns the port number actually allocated.

sin can be NULL, in which case *sin->sin_family* is implicitly taken to be **AF_INET**. However, in this case, **bindresvport()** has no way to return the port number actually allocated. (This information can later be obtained using **getsockname(2)**.)

RETURN VALUE

bindresvport() returns 0 on success; otherwise -1 is returned and *errno* set to indicate the cause of the error.

ERRORS

bindresvport() can fail for any of the same reasons as **bind(2)**. In addition, the following errors may occur:

EACCES

The caller did not have superuser privilege (to be precise: the **CAP_NET_BIND_SERVICE** capability is required).

EADDRINUSE

All privileged ports are in use.

EAFNOSUPPORT (EPFNOSUPPORT in glibc 2.7 and earlier)

sin is not NULL and *sin->sin_family* is not **AF_INET**.

ATTRIBUTES**Multithreading (see pthreads(7))**

Before glibc 2.17, the **bindresvport()** function uses a static variable that is not protected, so it is not thread-safe.

Since glibc 2.17, the **bindresvport()** function uses a lock to protect the static variable, so it is thread-safe.

CONFORMING TO

Not in POSIX.1-2001. Present on the BSDs, Solaris, and many other systems.

NOTES

Unlike some **bindresvport()** implementations, the glibc implementation ignores any value that the caller supplies in *sin->sin_port*.

SEE ALSO

bind(2), **getsockname(2)**

COLOPHON

This page is part of release 3.74 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <http://www.kernel.org/doc/man-pages/>.