NAME

gethostname, sethostname − get/set hostname

LIBRARY

Standard C library (*libc*, *−lc*)

SYNOPSIS

```
#include <unistd.h>

int gethostname(char *name, size_t len);
int sethostname(const char *name, size_t len);
```

Feature Test Macro Requirements for glibc (see **feature_test_macros**(7)):

gethostname():
_XOPEN_SOURCE >= 500 || _POSIX_C_SOURCE >= 200112L
|| /* glibc 2.19 and earlier */ _BSD_SOURCE

sethostname():
Since glibc 2.21:
_DEFAULT_SOURCE
In glibc 2.19 and 2.20:
_DEFAULT_SOURCE || (_XOPEN_SOURCE && _XOPEN_SOURCE < 500)
Up to and including glibc 2.19:
_BSD_SOURCE || (_XOPEN_SOURCE && _XOPEN_SOURCE < 500)

DESCRIPTION

These system calls are used to access or to change the system hostname. More precisely, they operate on the hostname associated with the calling process's UTS namespace.

sethostname() sets the hostname to the value given in the character array *name*. The*len* ar gument specifies the number of bytes in *name*. (Thus,*name* does not require a terminating null byte.)

gethostname() returns the null-terminated hostname in the character array *name*, which has a length of *len* bytes. If the null-terminated hostname is too large to fit, then the name is truncated, and no error is returned (but see NOTES below). POSIX.1 says that if such truncation occurs, then it is unspecified whether the returned buffer includes a terminating null byte.

RETURN VALUE

On success, zero is returned. On error, −1 is returned, and *errno* is set to indicate the error.

ERRORS

EFAULT
name is an invalid address.

EINVAL
len is negative or, for **sethostname**(), *len* is larger than the maximum allowed size.

ENAMETOOLONG
(glibc **gethostname**()) *len* is smaller than the actual size. (Before glibc 2.1, glibc uses **EINVAL** for this case.)

EPERM
For **sethostname**(), the caller did not have the **CAP_SYS_ADMIN** capability in the user namespace associated with its UTS namespace (see **namespaces**(7)).

STANDARDS

SVr4, 4.4BSD (these interfaces first appeared in 4.2BSD). POSIX.1-2001 and POSIX.1-2008 specify **gethostname**() but not **sethostname**().

NOTES

SUSv2 guarantees that "Host names are limited to 255 bytes". POSIX.1 guarantees that "Host names (not including the terminating null byte) are limited to **HOST_NAME_MAX** bytes". On Linux, **HOST_NAME_MAX** is defined with the value 64, which has been the limit since Linux 1.0 (earlier

kernels imposed a limit of 8 bytes).

C library/kernel differences

The GNU C library does not employ the **gethostname**() system call; instead, it implements **gethostname**() as a library function that calls **uname**(2) and copies up to *len* bytes from the returned *nodename* field into *name*. Having performed the copy, the function then checks if the length of the *nodename* was greater than or equal to *len*, and if it is, then the function returns −1 with *errno* set to **ENAMETOOLONG**; in this case, a terminating null byte is not included in the returned *name*.

Versions of glibc before glibc 2.2 handle the case where the length of the *nodename* was greater than or equal to *len* differently: nothing is copied into *name* and the function returns −1 with *errno* set to **ENAMETOOLONG**.

SEE ALSO

hostname(1), **getdomainname**(2), **setdomainname**(2), **uname**(2), **uts_namespaces**(7)