

**NAME**

getdomainname, setdomainname – get/set NIS domain name

**SYNOPSIS**

```
#include <unistd.h>
```

```
int getdomainname(char *name, size_t len);
int setdomainname(const char *name, size_t len);
```

Feature Test Macro Requirements for glibc (see [feature\\_test\\_macros\(7\)](#)):

**getdomainname()**, **setdomainname()**:

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 functions are used to access or to change the NIS domain name of the host system. More precisely, they operate on the NIS domain name associated with the calling process's UTS namespace.

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

**getdomainname()** returns the null-terminated domain name in the character array *name*, which has a length of *len* bytes. If the null-terminated domain name requires more than *len* bytes, **getdomainname()** returns the first *len* bytes (glibc) or gives an error (libc).

**RETURN VALUE**

On success, zero is returned. On error,  $-1$  is returned, and *errno* is set appropriately.

**ERRORS**

**setdomainname()** can fail with the following errors:

**EFAULT**

*name* pointed outside of user address space.

**EINVAL**

*len* was negative or too large.

**EPERM**

The caller did not have the **CAP\_SYS\_ADMIN** capability in the user namespace associated with its UTS namespace (see [namespaces\(7\)](#)).

**getdomainname()** can fail with the following errors:

**EINVAL**

For **getdomainname()** under libc: *name* is NULL or *name* is longer than *len* bytes.

**CONFORMING TO**

POSIX does not specify these calls.

**NOTES**

Since Linux 1.0, the limit on the length of a domain name, including the terminating null byte, is 64 bytes. In older kernels, it was 8 bytes.

On most Linux architectures (including x86), there is no **getdomainname()** system call; instead, glibc implements **getdomainname()** as a library function that returns a copy of the *domainname* field returned from a call to **uname(2)**.

**SEE ALSO**

[gethostname\(2\)](#), [sethostname\(2\)](#), [uname\(2\)](#), [uts\\_namespaces\(7\)](#)

**COLOPHON**

This page is part of release 5.05 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 <https://www.kernel.org/doc/man-pages/>.