

NAME

confstr - get configuration dependent string variables

SYNOPSIS

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

```
size_t confstr(int name, char *buf, size_t len);
```

Feature Test Macro Requirements for glibc (see [feature_test_macros\(7\)](#)):

```
confstr(): _POSIX_C_SOURCE >= 2 || _XOPEN_SOURCE
```

DESCRIPTION

confstr() gets the value of configuration-dependent string variables.

The *name* argument is the system variable to be queried. The following variables are supported:

_CS_GNU_LIBC_VERSION (GNU C library only; since glibc 2.3.2)

A string which identifies the GNU C library version on this system (e.g, glibc 2.3.4).

_CS_GNU_LIBPTHREAD_VERSION (GNU C library only; since glibc 2.3.2)

A string which identifies the POSIX implementation supplied by this C library (e.g, NPTL 2.3.4 or linuxthreads-0.10).

_CS_PATH

A value for the **PATH** variable which indicates where all the POSIX.2 standard utilities can be found.

If *buf* is not NULL and *len* is not zero, **confstr()** copies the value of the string to *buf* truncated to *len* - 1 bytes if necessary, with a null byte (0) as terminator. This can be detected by comparing the return value of **confstr()** against *len*.

If *len* is zero and *buf* is NULL, **confstr()** just returns the value as defined below.

RETURN VALUE

If *name* is a valid configuration variable, **confstr()** returns the number of bytes (including the terminating null byte) that would be required to hold the entire value of that variable. This value may be greater than *len*, which means that the value in *buf* is truncated.

If *name* is a valid configuration variable, but that variable does not have a value, then **confstr()** returns 0. If *name* does not correspond to a valid configuration variable, **confstr()** returns 0, and *errno* is set to **EINVAL**.

ERRORS

EINVAL

The value of *name* is invalid.

CONFORMING TO

POSIX.1-2001.

EXAMPLE

The following code fragment determines the path where to find the POSIX.2 system utilities:

```
char *pathbuf;
size_t n;

n = confstr(_CS_PATH, NULL, (size_t) 0);
pathbuf = malloc(n)
if (pathbuf == NULL)
abort();
confstr(_CS_PATH, pathbuf, n);
```

SEE ALSO

[getconf\(1\)](#), [sh\(1\)](#), [exec\(3\)](#), [fpathconf\(3\)](#) [sysconf\(3\)](#) [pathconf\(3\)](#) [system\(3\)](#)

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/>.