

NAME

clearenv – clear the environment

LIBRARY

Standard C library (*libc*, *-lc*)

SYNOPSIS

```
#include <stdlib.h>
```

```
int clearenv(void);
```

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

```
clearenv():
/* glibc >= 2.19: */ _DEFAULT_SOURCE
|| /* glibc <= 2.19: */ _SVID_SOURCE || _BSD_SOURCE
```

DESCRIPTION

The **clearenv()** function clears the environment of all name-value pairs and sets the value of the external variable *environ* to NULL. After this call, new variables can be added to the environment using **putenv(3)** and **setenv(3)**.

RETURN VALUE

The **clearenv()** function returns zero on success, and a nonzero value on failure.

ATTRIBUTES

For an explanation of the terms used in this section, see **attributes(7)**.

Interface	Attribute	Value
clearenv()	Thread safety	MT-Unsafe const:env

STANDARDS

putenv()
POSIX.1-2008.

clearenv()
None.

HISTORY

putenv()
glibc 2.0. POSIX.1-2001.

clearenv()
glibc 2.0.

Various UNIX variants (DG/UX, HP-UX, QNX, ...). POSIX.9 (bindings for FORTRAN77). POSIX.1-1996 did not accept **clearenv()** and **putenv(3)**, but changed its mind and scheduled these functions for some later issue of this standard (see §B.4.6.1). However, POSIX.1-2001 adds only **putenv(3)**, and rejected **clearenv()**.

NOTES

On systems where **clearenv()** is unavailable, the assignment

```
environ = NULL;
```

will probably do.

The **clearenv()** function may be useful in security-conscious applications that want to precisely control the environment that is passed to programs executed using **exec(3)**. The application would do this by first clearing the environment and then adding select environment variables.

Note that the main effect of **clearenv()** is to adjust the value of the pointer **environ(7)**; this function does not erase the contents of the buffers containing the environment definitions.

The DG/UX and Tru64 man pages write: If *environ* has been modified by anything other than the **putenv(3)**, **getenv(3)**, or **clearenv()** functions, then **clearenv()** will return an error and the process environment will remain unchanged.

SEE ALSO**getenv(3), putenv(3), setenv(3), unsetenv(3), environ(7)**