



Rocky Enterprise Linux 9.2 Manual Pages on command 'putenv.3'

C:\>man putenv.3

PUTENV(3) Linux Programmer's Manual PUTENV(3)

NAME

putenv - change or add an environment variable

SYNOPSIS

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

```
int putenv(char *string);
```

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

```
putenv(): _XOPEN_SOURCE
```

```
    || /* Glibc since 2.19: */ _DEFAULT_SOURCE
```

```
    || /* Glibc versions <= 2.19: */ _SVID_SOURCE
```

DESCRIPTION

The `putenv()` function adds or changes the value of environment variables. The argument `string` is of the form `name=value`. If `name` does not already exist in the environment, then `string` is added to the environment. If `name` does exist, then the value of `name` in the environment is changed to `value`. The `string` pointed to by `string` becomes part of the environment, so altering the string changes the environment.

RETURN VALUE

The `putenv()` function returns zero on success, or nonzero if an error occurs. In the event of an error, `errno` is set to indicate the cause.

ERRORS

ENOMEM Insufficient space to allocate new environment.

ATTRIBUTES

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

??

?Interface ? Attribute ? Value ?

??

?putenv() ? Thread safety ? MT-Unsafe const:env ?

??

CONFORMING TO

POSIX.1-2001, POSIX.1-2008, SVr4, 4.3BSD.

NOTES

The `putenv()` function is not required to be reentrant, and the one in glibc 2.0 is not, but the glibc 2.1 version is.

Since version 2.1.2, the glibc implementation conforms to SUSv2: the pointer string given to `putenv()` is used. In particular, this string becomes part of the environment; changing it later will change the environment. (Thus, it is an error to call `putenv()` with an automatic variable as the argument, then return from the calling function while string is still part of the environment.) However, glibc versions 2.0 to 2.1.1 differ: a copy of the string is used. On the one hand this causes a memory leak, and on the other hand it violates SUSv2.

The 4.4BSD version, like glibc 2.0, uses a copy.

SUSv2 removes the `const` from the prototype, and so does glibc 2.1.3.

The GNU C library implementation provides a nonstandard extension. If string does not include an equal sign:

```
putenv("NAME");
```

then the named variable is removed from the caller's environment.

SEE ALSO

`clearenv(3)`, `getenv(3)`, `setenv(3)`, `unsetenv(3)`, `environ(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/>.