



Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'setenv.3p' command

\$ man setenv.3p

SETENV(3P) POSIX Programmer's Manual SETENV(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

setenv ? add or change environment variable

SYNOPSIS

```
#include <stdlib.h>

int setenv(const char *envname, const char *envval, int overwrite);
```

DESCRIPTION

The setenv() function shall update or add a variable in the environment of the calling process. The envname argument points to a string containing the name of an environment variable to be added or altered. The environment variable shall be set to the value to which envval points. The function shall fail if envname points to a string which contains an '=' character. If the environment variable named by envname already exists and the value of overwrite is non-zero, the function shall return success and the environment shall be updated. If the environment variable named by envname already exists and the value of overwrite is zero, the function shall return success and the environment shall remain unchanged.

The `setenv()` function shall update the list of pointers to which `environ` points.

The strings described by `envname` and `envval` are copied by this function.

The `setenv()` function need not be thread-safe.

RETURN VALUE

Upon successful completion, zero shall be returned. Otherwise, -1 shall be returned, `errno` set to indicate the error, and the environment shall be unchanged.

ERRORS

The `setenv()` function shall fail if:

EINVAL The `envname` argument points to an empty string or points to a string containing an '=' character.

ENOMEM Insufficient memory was available to add a variable or its value to the environment.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

See `exec()` for restrictions on changing the environment in multithreaded applications.

RATIONALE

Unanticipated results may occur if `setenv()` changes the external variable `environ`. In particular, if the optional `envp` argument to `main()` is present, it is not changed, and thus may point to an obsolete copy of the environment (as may any other copy of `environ`). However, other than the aforementioned restriction, the standard developers intended that the traditional method of walking through the environment by way of the `environ` pointer must be supported.

It was decided that `setenv()` should be required by this version because it addresses a piece of missing functionality, and does not impose a significant burden on the implementor.

There was considerable debate as to whether the System V `putenv()` func?

tion or the BSD `setenv()` function should be required as a mandatory function. The `setenv()` function was chosen because it permitted the implementation of the `unsetenv()` function to delete environmental variables, without specifying an additional interface. The `putenv()` function is available as part of the XSI option.

The standard developers considered requiring that `setenv()` indicate an error when a call to it would result in exceeding `{ARG_MAX}`. The requirement was rejected since the condition might be temporary, with the application eventually reducing the environment size. The ultimate success or failure depends on the size at the time of a call to `exec`, which returns an indication of this error condition.

See also the RATIONALE section in `getenv()`.

FUTURE DIRECTIONS

None.

SEE ALSO

`exec`, `getenv()`, `putenv()`, `unsetenv()`

The Base Definitions volume of POSIX.1-2017, `<stdlib.h>`, `<sys_types.h>`, `<unistd.h>`

COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.opengroup.org/unix/online.html>.

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see https://www.kernel.org/doc/man-pages/reporting_bugs.html.