



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 'limits.h.0p' command

\$ man limits.h.0p

limits.h(0P) POSIX Programmer's Manual limits.h(0P)

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

limits.h ? implementation-defined constants

SYNOPSIS

```
#include <limits.h>
```

DESCRIPTION

Some of the functionality described on this reference page extends the ISO C standard. Applications shall define the appropriate feature test macro (see the System Interfaces volume of POSIX.1?2017, Section 2.2, The Compilation Environment) to enable the visibility of these symbols in this header.

Many of the symbols listed here are not defined by the ISO/IEC 9899:1999 standard. Such symbols are not shown as CX shaded, except under the heading ``Numerical Limits''.

The <limits.h> header shall define macros and symbolic constants for various limits. Different categories of limits are described below, representing various limits on resources that the implementation imposes on applications. All macros and symbolic constants defined in

this header shall be suitable for use in #if preprocessing directives.

Implementations may choose any appropriate value for each limit, provided it is not more restrictive than the Minimum Acceptable Values listed below. Symbolic constant names beginning with `_POSIX` may be found in `<unistd.h>`.

Applications should not assume any particular value for a limit. To achieve maximum portability, an application should not require more resource than the Minimum Acceptable Value quantity. However, an application wishing to avail itself of the full amount of a resource available on an implementation may make use of the value given in `<limits.h>` on that particular implementation, by using the macros and symbolic constants listed below. It should be noted, however, that many of the listed limits are not invariant, and at runtime, the value of the limit may differ from those given in this header, for the following reasons:

- * The limit is pathname-dependent.
- * The limit differs between the compile and runtime machines.

For these reasons, an application may use the `fpathconf()`, `pathconf()`, and `sysconf()` functions to determine the actual value of a limit at runtime.

The items in the list ending in `_MIN` give the most negative values that the mathematical types are guaranteed to be capable of representing. Numbers of a more negative value may be supported on some implementations, as indicated by the `<limits.h>` header on the implementation, but applications requiring such numbers are not guaranteed to be portable to all implementations. For positive constants ending in `_MIN`, this indicates the minimum acceptable value.

Runtime Invariant Values (Possibly Indeterminate)

A definition of one of the symbolic constants in the following list shall be omitted from `<limits.h>` on specific implementations where the corresponding value is equal to or greater than the stated minimum, but is unspecified.

This indetermination might depend on the amount of available memory space on a specific instance of a specific implementation. The actual

value supported by a specific instance shall be provided by the `sysconf()` function.

{AIO_LISTIO_MAX}

Maximum number of I/O operations in a single list I/O call supported by the implementation.

Minimum Acceptable Value: {_POSIX_AIO_LISTIO_MAX}

{AIO_MAX}

Maximum number of outstanding asynchronous I/O operations supported by the implementation.

Minimum Acceptable Value: {_POSIX_AIO_MAX}

{AIO_PRIO_DELTA_MAX}

The maximum amount by which a process can decrease its asynchronous I/O priority level from its own scheduling priority.

Minimum Acceptable Value: 0

{ARG_MAX}

Maximum length of argument to the exec functions including environment data.

Minimum Acceptable Value: {_POSIX_ARG_MAX}

{ATEXIT_MAX}

Maximum number of functions that may be registered with `atexit()`.

Minimum Acceptable Value: 32

{CHILD_MAX}

Maximum number of simultaneous processes per real user ID.

Minimum Acceptable Value: {_POSIX_CHILD_MAX}

{DELAYTIMER_MAX}

Maximum number of timer expiration overruns.

Minimum Acceptable Value: {_POSIX_DELAYTIMER_MAX}

{HOST_NAME_MAX}

Maximum length of a host name (not including the terminating null) as returned from the `gethostname()` function.

Minimum Acceptable Value: {_POSIX_HOST_NAME_MAX}

{IOV_MAX}

Maximum number of iovec structures that one process has available

for use with `readv()` or `writev()`.

Minimum Acceptable Value: `{_XOPEN_IOV_MAX}`

`{LOGIN_NAME_MAX}`

Maximum length of a login name.

Minimum Acceptable Value: `{_POSIX_LOGIN_NAME_MAX}`

`{MQ_OPEN_MAX}`

The maximum number of open message queue descriptors a process may hold.

Minimum Acceptable Value: `{_POSIX_MQ_OPEN_MAX}`

`{MQ_PRIO_MAX}`

The maximum number of message priorities supported by the implementation.

Minimum Acceptable Value: `{_POSIX_MQ_PRIO_MAX}`

`{OPEN_MAX}`

A value one greater than the maximum value that the system may assign to a newly-created file descriptor.

Minimum Acceptable Value: `{_POSIX_OPEN_MAX}`

`{PAGESIZE}`

Size in bytes of a page.

Minimum Acceptable Value: 1

`{PAGE_SIZE}`

Equivalent to `{PAGESIZE}`. If either `{PAGESIZE}` or `{PAGE_SIZE}` is defined, the other is defined with the same value.

`{PTHREAD_DESTRUCTOR_ITERATIONS}`

Maximum number of attempts made to destroy a thread's thread-specific data values on thread exit.

Minimum Acceptable Value: `{_POSIX_THREAD_DESTRUCTOR_ITERATIONS}`

`{PTHREAD_KEYS_MAX}`

Maximum number of data keys that can be created by a process.

Minimum Acceptable Value: `{_POSIX_THREAD_KEYS_MAX}`

`{PTHREAD_STACK_MIN}`

Minimum size in bytes of thread stack storage.

Minimum Acceptable Value: 0

{PTHREAD_THREADS_MAX}

Maximum number of threads that can be created per process.

Minimum Acceptable Value: {_POSIX_THREAD_THREADS_MAX}

{RTSIG_MAX}

Maximum number of realtime signals reserved for application use in this implementation.

Minimum Acceptable Value: {_POSIX_RTSIG_MAX}

{SEM_NSEMS_MAX}

Maximum number of semaphores that a process may have.

Minimum Acceptable Value: {_POSIX_SEM_NSEMS_MAX}

{SEM_VALUE_MAX}

The maximum value a semaphore may have.

Minimum Acceptable Value: {_POSIX_SEM_VALUE_MAX}

{SIGQUEUE_MAX}

Maximum number of queued signals that a process may send and have pending at the receiver(s) at any time.

Minimum Acceptable Value: {_POSIX_SIGQUEUE_MAX}

{SS_REPL_MAX}

The maximum number of replenishment operations that may be simultaneously pending for a particular sporadic server scheduler.

Minimum Acceptable Value: {_POSIX_SS_REPL_MAX}

{STREAM_MAX}

Maximum number of streams that one process can have open at one time. If defined, it has the same value as {FOPEN_MAX} (see <stdio.h>).

Minimum Acceptable Value: {_POSIX_STREAM_MAX}

{SYMLOOP_MAX}

Maximum number of symbolic links that can be reliably traversed in the resolution of a pathname in the absence of a loop.

Minimum Acceptable Value: {_POSIX_SYMLOOP_MAX}

{TIMER_MAX}

Maximum number of timers per process supported by the implementation.

Minimum Acceptable Value: `{_POSIX_TIMER_MAX}`

`{TRACE_EVENT_NAME_MAX}`

Maximum length of the trace event name (not including the terminating null).

Minimum Acceptable Value: `{_POSIX_TRACE_EVENT_NAME_MAX}`

`{TRACE_NAME_MAX}`

Maximum length of the trace generation version string or of the trace stream name (not including the terminating null).

Minimum Acceptable Value: `{_POSIX_TRACE_NAME_MAX}`

`{TRACE_SYS_MAX}`

Maximum number of trace streams that may simultaneously exist in the system.

Minimum Acceptable Value: `{_POSIX_TRACE_SYS_MAX}`

`{TRACE_USER_EVENT_MAX}`

Maximum number of user trace event type identifiers that may simultaneously exist in a traced process, including the predefined user trace event `POSIX_TRACE_UNNAMED_USER_EVENT`.

Minimum Acceptable Value: `{_POSIX_TRACE_USER_EVENT_MAX}`

`{TTY_NAME_MAX}`

Maximum length of terminal device name.

Minimum Acceptable Value: `{_POSIX_TTY_NAME_MAX}`

`{TZNAME_MAX}`

Maximum number of bytes supported for the name of a timezone (not of the `TZ` variable).

Minimum Acceptable Value: `{_POSIX_TZNAME_MAX}`

Note: The length given by `{TZNAME_MAX}` does not include the quoting characters mentioned in Section 8.3, Other Environment Variables.

Pathname Variable Values

The values in the following list may be constants within an implementation or may vary from one pathname to another. For example, file systems or directories may have different characteristics.

A definition of one of the symbolic constants in the following list

shall be omitted from the <limits.h> header on specific implementations where the corresponding value is equal to or greater than the stated minimum, but where the value can vary depending on the file to which it is applied. The actual value supported for a specific pathname shall be provided by the pathconf() function.

{FILESIZEBITS}

Minimum number of bits needed to represent, as a signed integer value, the maximum size of a regular file allowed in the specified directory.

Minimum Acceptable Value: 32

{LINK_MAX}

Maximum number of links to a single file.

Minimum Acceptable Value: {_POSIX_LINK_MAX}

{MAX_CANON}

Maximum number of bytes in a terminal canonical input line.

Minimum Acceptable Value: {_POSIX_MAX_CANON}

{MAX_INPUT}

Minimum number of bytes for which space is available in a terminal input queue; therefore, the maximum number of bytes a conforming application may require to be typed as input before reading them.

Minimum Acceptable Value: {_POSIX_MAX_INPUT}

{NAME_MAX}

Maximum number of bytes in a filename (not including the terminating null of a filename string).

Minimum Acceptable Value: {_POSIX_NAME_MAX}

Minimum Acceptable Value: {_XOPEN_NAME_MAX}

{PATH_MAX}

Maximum number of bytes the implementation will store as a pathname in a user-supplied buffer of unspecified size, including the terminating null character. Minimum number the implementation will accept as the maximum number of bytes in a pathname.

Minimum Acceptable Value: {_POSIX_PATH_MAX}

Minimum Acceptable Value: {_XOPEN_PATH_MAX}

{PIPE_BUF}

Maximum number of bytes that is guaranteed to be atomic when writing to a pipe.

Minimum Acceptable Value: {_POSIX_PIPE_BUF}

{POSIX_ALLOC_SIZE_MIN}

Minimum number of bytes of storage actually allocated for any portion of a file.

Minimum Acceptable Value: Not specified.

{POSIX_REC_INCR_XFER_SIZE}

Recommended increment for file transfer sizes between the {POSIX_REC_MIN_XFER_SIZE} and {POSIX_REC_MAX_XFER_SIZE} values.

Minimum Acceptable Value: Not specified.

{POSIX_REC_MAX_XFER_SIZE}

Maximum recommended file transfer size.

Minimum Acceptable Value: Not specified.

{POSIX_REC_MIN_XFER_SIZE}

Minimum recommended file transfer size.

Minimum Acceptable Value: Not specified.

{POSIX_REC_XFER_ALIGN}

Recommended file transfer buffer alignment.

Minimum Acceptable Value: Not specified.

{SYMLINK_MAX}

Maximum number of bytes in a symbolic link.

Minimum Acceptable Value: {_POSIX_SYMLINK_MAX}

Runtime Inerasable Values

The magnitude limitations in the following list shall be fixed by specific implementations. An application should assume that the value of the symbolic constant defined by <limits.h> in a specific implementation is the minimum that pertains whenever the application is run under that implementation. A specific instance of a specific implementation may increase the value relative to that supplied by <limits.h> for that implementation. The actual value supported by a specific instance shall

be provided by the `sysconf()` function.

{BC_BASE_MAX}

Maximum obase values allowed by the `bc` utility.

Minimum Acceptable Value: {_POSIX2_BC_BASE_MAX}

{BC_DIM_MAX}

Maximum number of elements permitted in an array by the `bc` utility.

Minimum Acceptable Value: {_POSIX2_BC_DIM_MAX}

{BC_SCALE_MAX}

Maximum scale value allowed by the `bc` utility.

Minimum Acceptable Value: {_POSIX2_BC_SCALE_MAX}

{BC_STRING_MAX}

Maximum length of a string constant accepted by the `bc` utility.

Minimum Acceptable Value: {_POSIX2_BC_STRING_MAX}

{CHARCLASS_NAME_MAX}

Maximum number of bytes in a character class name.

Minimum Acceptable Value: {_POSIX2_CHARCLASS_NAME_MAX}

{COLL_WEIGHTS_MAX}

Maximum number of weights that can be assigned to an entry of the `LC_COLLATE` order keyword in the locale definition file; see Chapter 7, Locale.

Minimum Acceptable Value: {_POSIX2_COLL_WEIGHTS_MAX}

{EXPR_NEST_MAX}

Maximum number of expressions that can be nested within parentheses by the `expr` utility.

Minimum Acceptable Value: {_POSIX2_EXPR_NEST_MAX}

{LINE_MAX}

Unless otherwise noted, the maximum length, in bytes, of a utility's input line (either standard input or another file), when the utility is described as processing text files. The length includes room for the trailing <newline>.

Minimum Acceptable Value: {_POSIX2_LINE_MAX}

{NGROUPS_MAX}

Maximum number of simultaneous supplementary group IDs per process.

Minimum Acceptable Value: `{_POSIX_NGROUPS_MAX}`

`{RE_DUP_MAX}`

Maximum number of repeated occurrences of a BRE or ERE interval expression; see Section 9.3.6, BREs Matching Multiple Characters and Section 9.4.6, EREs Matching Multiple Characters.

Minimum Acceptable Value: `{_POSIX_RE_DUP_MAX}`

Maximum Values

The `<limits.h>` header shall define the following symbolic constants with the values shown. These are the most restrictive values for certain features on an implementation. A conforming implementation shall provide values no larger than these values. A conforming application must not require a smaller value for correct operation.

`{_POSIX_CLOCKRES_MIN}`

The resolution of the `CLOCK_REALTIME` clock, in nanoseconds.

Value: 20 000 000

If the Monotonic Clock option is supported, the resolution of the `CLOCK_MONOTONIC` clock, in nanoseconds, is represented by `{_POSIX_CLOCKRES_MIN}`.

Minimum Values

The `<limits.h>` header shall define the following symbolic constants with the values shown. These are the most restrictive values for certain features on an implementation conforming to this volume of POSIX.1?2017. Related symbolic constants are defined elsewhere in this volume of POSIX.1?2017 which reflect the actual implementation and which need not be as restrictive. For each of these limits, a conforming implementation shall provide a value at least this large or shall have no limit. A strictly conforming application must not require a larger value for correct operation.

`{_POSIX_AIO_LISTIO_MAX}`

The number of I/O operations that can be specified in a list I/O call.

Value: 2

{_POSIX_AIO_MAX}

The number of outstanding asynchronous I/O operations.

Value: 1

{_POSIX_ARG_MAX}

Maximum length of argument to the exec functions including environment data.

Value: 4 096

{_POSIX_CHILD_MAX}

Maximum number of simultaneous processes per real user ID.

Value: 25

{_POSIX_DELAYTIMER_MAX}

The number of timer expiration overruns.

Value: 32

{_POSIX_HOST_NAME_MAX}

Maximum length of a host name (not including the terminating null) as returned from the gethostname() function.

Value: 255

{_POSIX_LINK_MAX}

Maximum number of links to a single file.

Value: 8

{_POSIX_LOGIN_NAME_MAX}

The size of the storage required for a login name, in bytes (including the terminating null).

Value: 9

{_POSIX_MAX_CANON}

Maximum number of bytes in a terminal canonical input queue.

Value: 255

{_POSIX_MAX_INPUT}

Maximum number of bytes allowed in a terminal input queue.

Value: 255

{_POSIX_MQ_OPEN_MAX}

The number of message queues that can be open for a single

process.

Value: 8

{_POSIX_MQ_PRIO_MAX}

The maximum number of message priorities supported by the implementation.

Value: 32

{_POSIX_NAME_MAX}

Maximum number of bytes in a filename (not including the terminating null of a filename string).

Value: 14

{_POSIX_NGROUPS_MAX}

Maximum number of simultaneous supplementary group IDs per process.

Value: 8

{_POSIX_OPEN_MAX}

A value one greater than the maximum value that the system may assign to a newly-created file descriptor.

Value: 20

{_POSIX_PATH_MAX}

Minimum number the implementation will accept as the maximum number of bytes in a pathname.

Value: 256

{_POSIX_PIPE_BUF}

Maximum number of bytes that is guaranteed to be atomic when writing to a pipe.

Value: 512

{_POSIX_RE_DUP_MAX}

Maximum number of repeated occurrences of a BRE or ERE interval expression; see Section 9.3.6, BREs Matching Multiple Characters and Section 9.4.6, EREs Matching Multiple Characters.

Value: 255

{_POSIX_RTSIG_MAX}

The number of realtime signal numbers reserved for application

use.

Value: 8

{_POSIX_SEM_NSEMS_MAX}

The number of semaphores that a process may have.

Value: 256

{_POSIX_SEM_VALUE_MAX}

The maximum value a semaphore may have.

Value: 32 767

{_POSIX_SIGQUEUE_MAX}

The number of queued signals that a process may send and have pending at the receiver(s) at any time.

Value: 32

{_POSIX_SSIZE_MAX}

The value that can be stored in an object of type `ssize_t`.

Value: 32 767

{_POSIX_SS_REPL_MAX}

The number of replenishment operations that may be simultaneously pending for a particular sporadic server scheduler.

Value: 4

{_POSIX_STREAM_MAX}

The number of streams that one process can have open at one time.

Value: 8

{_POSIX_SYMLINK_MAX}

The number of bytes in a symbolic link.

Value: 255

{_POSIX_SYMLINK_MAX}

The number of symbolic links that can be traversed in the resolution of a pathname in the absence of a loop.

Value: 8

{_POSIX_THREAD_DESTRUCTOR_ITERATIONS}

The number of attempts made to destroy a thread's thread-specific data values on thread exit.

Value: 4

{_POSIX_THREAD_KEYS_MAX}

The number of data keys per process.

Value: 128

{_POSIX_THREAD_THREADS_MAX}

The number of threads per process.

Value: 64

{_POSIX_TIMER_MAX}

The per-process number of timers.

Value: 32

{_POSIX_TRACE_EVENT_NAME_MAX}

The length in bytes of a trace event name (not including the terminating null).

Value: 30

{_POSIX_TRACE_NAME_MAX}

The length in bytes of a trace generation version string or a trace stream name (not including the terminating null).

Value: 8

{_POSIX_TRACE_SYS_MAX}

The number of trace streams that may simultaneously exist in the system.

Value: 8

{_POSIX_TRACE_USER_EVENT_MAX}

The number of user trace event type identifiers that may simultaneously exist in a traced process, including the predefined user trace event `POSIX_TRACE_UNNAMED_USER_EVENT`.

Value: 32

{_POSIX_TTY_NAME_MAX}

The size of the storage required for a terminal device name, in bytes (including the terminating null).

Value: 9

{_POSIX_TZNAME_MAX}

Maximum number of bytes supported for the name of a timezone (not of the TZ variable).

Value: 6

Note: The length given by `{_POSIX_TZNAME_MAX}` does not include the quoting characters mentioned in Section 8.3, Other Environment Variables.

`{_POSIX2_BC_BASE_MAX}`

Maximum obase values allowed by the `bc` utility.

Value: 99

`{_POSIX2_BC_DIM_MAX}`

Maximum number of elements permitted in an array by the `bc` utility.

Value: 2048

`{_POSIX2_BC_SCALE_MAX}`

Maximum scale value allowed by the `bc` utility.

Value: 99

`{_POSIX2_BC_STRING_MAX}`

Maximum length of a string constant accepted by the `bc` utility.

Value: 1000

`{_POSIX2_CHARCLASS_NAME_MAX}`

Maximum number of bytes in a character class name.

Value: 14

`{_POSIX2_COLL_WEIGHTS_MAX}`

Maximum number of weights that can be assigned to an entry of the `LC_COLLATE` order keyword in the locale definition file; see Chapter 7, Locale.

Value: 2

`{_POSIX2_EXPR_NEST_MAX}`

Maximum number of expressions that can be nested within parentheses by the `expr` utility.

Value: 32

`{_POSIX2_LINE_MAX}`

Unless otherwise noted, the maximum length, in bytes, of a utility's input line (either standard input or another file), when the utility is described as processing text files. The length in?

cludes room for the trailing <newline>.

Value: 2 048

{_POSIX2_RE_DUP_MAX}

Maximum number of repeated occurrences of a BRE or ERE interval expression; see Section 9.3.6, BREs Matching Multiple Characters and Section 9.4.6, EREs Matching Multiple Characters.

Value: 255

{_XOPEN_IOV_MAX}

Maximum number of iovec structures that one process has available for use with readv() or writev().

Value: 16

{_XOPEN_NAME_MAX}

Maximum number of bytes in a filename (not including the terminating null of a filename string).

Value: 255

{_XOPEN_PATH_MAX}

Minimum number the implementation will accept as the maximum number of bytes in a pathname.

Value: 1024

Numerical Limits

The <limits.h> header shall define the following macros and, except for {CHAR_BIT}, {LONG_BIT}, {MB_LEN_MAX}, and {WORD_BIT}, they shall be replaced by expressions that have the same type as would an expression that is an object of the corresponding type converted according to the integer promotions.

If the value of an object of type char is treated as a signed integer when used in an expression, the value of {CHAR_MIN} is the same as that of {SCHAR_MIN} and the value of {CHAR_MAX} is the same as that of {SCHAR_MAX}. Otherwise, the value of {CHAR_MIN} is 0 and the value of {CHAR_MAX} is the same as that of {UCHAR_MAX}.

{CHAR_BIT}

Number of bits in a type char.

Value: 8

{CHAR_MAX}

Maximum value for an object of type char.

Value: {UCHAR_MAX} or {SCHAR_MAX}

{CHAR_MIN}

Minimum value for an object of type char.

Value: {SCHAR_MIN} or 0

{INT_MAX}

Maximum value for an object of type int.

Minimum Acceptable Value: 2 147 483 647

{INT_MIN}

Minimum value for an object of type int.

Maximum Acceptable Value: -2 147 483 647

{LLONG_MAX}

Maximum value for an object of type long long.

Minimum Acceptable Value: +9223372036854775807

{LLONG_MIN}

Minimum value for an object of type long long.

Maximum Acceptable Value: -9223372036854775807

{LONG_BIT}

Number of bits in an object of type long.

Minimum Acceptable Value: 32

{LONG_MAX}

Maximum value for an object of type long.

Minimum Acceptable Value: +2 147 483 647

{LONG_MIN}

Minimum value for an object of type long.

Maximum Acceptable Value: -2 147 483 647

{MB_LEN_MAX}

Maximum number of bytes in a character, for any supported locale.

Minimum Acceptable Value: 1

{SCHAR_MAX}

Maximum value for an object of type signed char.

Value: +127

{SCHAR_MIN}

Minimum value for an object of type signed char.

Value: -128

{SHRT_MAX}

Maximum value for an object of type short.

Minimum Acceptable Value: +32 767

{SHRT_MIN}

Minimum value for an object of type short.

Maximum Acceptable Value: -32 767

{SSIZE_MAX}

Maximum value for an object of type ssize_t.

Minimum Acceptable Value: {_POSIX_SSIZE_MAX}

{UCHAR_MAX}

Maximum value for an object of type unsigned char.

Value: 255

{UINT_MAX}

Maximum value for an object of type unsigned.

Minimum Acceptable Value: 4 294 967 295

{ULLONG_MAX}

Maximum value for an object of type unsigned long long.

Minimum Acceptable Value: 18446744073709551615

{ULONG_MAX}

Maximum value for an object of type unsigned long.

Minimum Acceptable Value: 4 294 967 295

{USHRT_MAX}

Maximum value for an object of type unsigned short.

Minimum Acceptable Value: 65 535

{WORD_BIT}

Number of bits in an object of type int.

Minimum Acceptable Value: 32

Other Invariant Values

The <limits.h> header shall define the following symbolic constants:

{NL_ARGMAX}

Maximum value of n in conversion specifications using the "%n\$" sequence in calls to the printf() and scanf() families of functions.

Minimum Acceptable Value: 9

{NL_LANGMAX}

Maximum number of bytes in a LANG name.

Minimum Acceptable Value: 14

{NL_MSGMAX}

Maximum message number.

Minimum Acceptable Value: 32 767

{NL_SETMAX}

Maximum set number.

Minimum Acceptable Value: 255

{NL_TEXTMAX}

Maximum number of bytes in a message string.

Minimum Acceptable Value: {_POSIX2_LINE_MAX}

{NZERO}

Default process priority.

Minimum Acceptable Value: 20

The following sections are informative.

APPLICATION USAGE

None.

RATIONALE

A request was made to reduce the value of {_POSIX_LINK_MAX} from the value of 8 specified for it in the POSIX.1?1990 standard to 2. The standard developers decided to deny this request for several reasons:

- * They wanted to avoid making any changes to the standard that could break conforming applications, and the requested change could have that effect.
- * The use of multiple hard links to a file cannot always be replaced with use of symbolic links. Symbolic links are semantically different from hard links in that they associate a pathname with another pathname rather than a pathname with a file. This has implications

for access control, file permanence, and transparency.

- * The original standard developers had considered the issue of allowing for implementations that did not in general support hard links, and decided that this would reduce consensus on the standard.

Systems that support historical versions of the development option of the ISO POSIX?2 standard retain the name `{_POSIX2_RE_DUP_MAX}` as an alias for `{_POSIX_RE_DUP_MAX}`.

`{PATH_MAX}`

IEEE PASC Interpretation 1003.1 #15 addressed the inconsistency in the standard with the definition of pathname and the description of `{PATH_MAX}`, allowing application developers to allocate either `{PATH_MAX}` or `{PATH_MAX}+1` bytes. The inconsistency has been removed by correction to the `{PATH_MAX}` definition to include the null character. With this change, applications that previously allocated `{PATH_MAX}` bytes will continue to succeed.

`{SYMLINK_MAX}`

This symbol refers to space for data that is stored in the file system, as opposed to `{PATH_MAX}` which is the length of a name that can be passed to a function. In some existing implementations, the pathnames pointed to by symbolic links are stored in the inodes of the links, so it is important that `{SYMLINK_MAX}` not be constrained to be as large as `{PATH_MAX}`.

FUTURE DIRECTIONS

None.

SEE ALSO

Chapter 7, Locale, `<stdio.h>`, `<unistd.h>`

The System Interfaces volume of POSIX.1?2017, Section 2.2, The Compilation Environment, `fpathconf()`, `sysconf()`

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 .

IEEE/The Open Group

2017

limits.h(OP)