man7.org > Linux > man-pages Linux/UNIX system programming training

sigemptyset(3p) — Linux manual page

PROLOG | NAME | SYNOPSIS | DESCRIPTION | RETURN VALUE | ERRORS | EXAMPLES | APPLICATION USAGE | RATIONALE | FUTURE DIRECTIONS | SEE ALSO | COPYRIGHT

Search online pages

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

PROLOG top

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 top

sigemptyset — initialize and empty a signal set

SYNOPSIS top

```
#include <signal.h>

int sigemptyset(sigset_t *set);
```

DESCRIPTION top

The *sigemptyset*() function initializes the signal set pointed to by *set*, such that all signals defined in POSIX.1-2008 are excluded.

RETURN VALUE top

Upon successful completion, *sigemptyset*() shall return 0; otherwise, it shall return -1 and set *errno* to indicate the error.

ERRORS top

No errors are defined.

The following sections are informative.

EXAMPLES top

None.

APPLICATION USAGE top

None.

RATIONALE top

The implementation of the *sigemptyset*() (or *sigfillset*()) function could quite trivially clear (or set) all the bits in the signal set. Alternatively, it would be reasonable to initialize part of the structure, such as a version field, to permit binary-compatibility between releases where the size of the set varies. For such reasons, either *sigemptyset*() or *sigfillset*() must be called prior to any other use of the signal set, even if such use is read-only (for example, as an argument to *sigpending*()). This function is not intended for dynamic allocation.

The *sigfillset*() and *sigemptyset*() functions require that the resulting signal set include (or exclude) all the signals defined in this volume of POSIX.1-2017. Although it is outside the scope of this volume of POSIX.1-2017 to place this requirement on signals that are implemented as extensions, it is recommended that implementation-defined signals also be affected by these functions. However, there may be a good reason for a particular signal not to be affected. For example, blocking or ignoring an implementation-defined signal may have undesirable side-effects, whereas the default action for that signal is harmless. In such a case, it would be preferable for such a signal to be excluded from the signal set returned by *sigfillset*().

In early proposals there was no distinction between invalid and unsupported signals (the names of optional signals that were not supported by an implementation were not defined by that implementation). The **[EINVAL]** error was thus specified as a required error for invalid signals. With that distinction, it is not necessary to require implementations of these functions to determine whether an optional signal is actually supported, as that could have a significant performance impact for little value. The error could have been required for invalid signals and optional for unsupported signals, but this seemed unnecessarily complex. Thus, the error is optional in both cases.

FUTURE DIRECTIONS top

None.

SEE ALSO top

Section 2.4, *Signal Concepts*, pthread_sigmask(3p), sigaction(3p), sigaddset(3p), sigdelset(3p), sigfillset(3p), sigismember(3p), sigpending(3p), sigsuspend(3p)

The Base Definitions volume of POSIX.1-2017, signal.h(0p)

COPYRIGHT top

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 SIGEMPTYSET(3P)

Pages that refer to this page: signal.h(0p), pthread_sigmask(3p), sigaction(3p), sigaddset(3p), sigdelset(3p), sigfillset(3p), sigismember(3p), sigpending(3p), sigsuspend(3p)

HTML rendering created 2021-08-27 by Michael Kerrisk, author of *The Linux Programming Interface*, maintainer of the Linux *man-pages* project.

For details of in-depth **Linux/UNIX system programming training courses** that I teach, look here.

Hosting by jambit GmbH.