



Rocky Enterprise Linux 9.2 Manual Pages on command 'sigevent.7'

C:\>man sigevent.7

SIGEVENT(7) Linux Programmer's Manual SIGEVENT(7)

NAME

sigevent - structure for notification from asynchronous routines

SYNOPSIS

```
#include <signal.h>

union signal {      /* Data passed with notification */
    int    sival_int;    /* Integer value */
    void  *sival_ptr;    /* Pointer value */
};

struct sigevent {

    int    sigev_notify; /* Notification method */

    int    sigev_signo; /* Notification signal */

    union signal sigev_value; /* Data passed with
                               notification */

    void    (*sigev_notify_function) (union signal);
            /* Function used for thread
               notification (SIGEV_THREAD) */

    void    *sigev_notify_attributes;
            /* Attributes for notification thread
               (SIGEV_THREAD) */

    pid_t   sigev_notify_thread_id;
            /* ID of thread to signal (SIGEV_THREAD_ID) */
```

};

DESCRIPTION

The `sigevent` structure is used by various APIs to describe the way a process is to be notified about an event (e.g., completion of an asynchronous request, expiration of a timer, or the arrival of a message).

The definition shown in the SYNOPSIS is approximate: some of the fields in the `sigevent` structure may be defined as part of a union. Programs should employ only those fields relevant to the value specified in `sigev_notify`.

The `sigev_notify` field specifies how notification is to be performed. This field can have one of the following values:

SIGEV_NONE

A "null" notification: don't do anything when the event occurs.

SIGEV_SIGNAL

Notify the process by sending the signal specified in `sigev_signo`.

If the signal is caught with a signal handler that was registered using the `sigaction(2)` `SA_SIGINFO` flag, then the following fields are set in the `siginfo_t` structure that is passed as the second argument of the handler:

`si_code` This field is set to a value that depends on the API delivering the notification.

`si_signo` This field is set to the signal number (i.e., the same value as in `sigev_signo`).

`si_value` This field is set to the value specified in `sigev_value`.

Depending on the API, other fields may also be set in the `siginfo_t` structure.

The same information is also available if the signal is accepted using `sigwaitinfo(2)`.

SIGEV_THREAD

Notify the process by invoking `sigev_notify_function` "as if" it were the start function of a new thread. (Among the implementation possibilities here are that each timer notification could result in the creation of a new thread, or that a single thread is created to receive all notifications.)

The function is invoked with `sigev_value` as its sole argument. If `sigev_notify_attributes` is not `NULL`, it should point to a `pthread_attr_t`

structure that defines attributes for the new thread (see `pthread_attr_init(3)`).

`SIGEV_THREAD_ID` (Linux-specific)

Currently used only by POSIX timers; see `timer_create(2)`.

SEE ALSO

`timer_create(2)`, `aio_fsync(3)`, `aio_read(3)`, `aio_write(3)`, `getaddrinfo_a(3)`,
`lio_listio(3)`, `mq_notify(3)`, `aio(7)`, `pthreads(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/>.

GNU

2017-07-13

SIGEVENT(7)