

**NAME**

`inotify_init`, `inotify_init1` - initialize an inotify instance

**SYNOPSIS**

```
#include <sys/inotify.h>
```

```
int inotify_init(void);
int inotify_init1(int flags);
```

**DESCRIPTION**

For an overview of the inotify API, see [inotify\(7\)](#).

`inotify_init()` initializes a new inotify instance and returns a file descriptor associated with a new inotify event queue.

If *flags* is 0, then `inotify_init1()` is the same as `inotify_init()`. The following values can be bit-wise ORed in *flags* to obtain different behavior:

**IN\_NONBLOCK**

Set the `O_NONBLOCK` file status flag on the new open file description. Using this flag saves extra calls to [fcntl\(2\)](#) to achieve the same result.

**IN\_CLOEXEC**

Set the close-on-exec (`FD_CLOEXEC`) flag on the new file descriptor. See the description of the `O_CLOEXEC` flag in [open\(2\)](#) for reasons why this may be useful.

**RETURN VALUE**

On success, these system calls return a new file descriptor. On error, -1 is returned, and *errno* is set to indicate the error.

**ERRORS****EINVAL**

(`inotify_init1()`) An invalid value was specified in *flags*.

**EMFILE**

The user limit on the total number of inotify instances has been reached.

**ENFILE**

The system limit on the total number of file descriptors has been reached.

**ENOMEM**

Insufficient kernel memory is available.

**VERSIONS**

`inotify_init()` first appeared in Linux 2.6.13; library support was added to glibc in version 2.4.

`inotify_init1()` was added in Linux 2.6.27; library support was added to glibc in version 2.9.

**CONFORMING TO**

These system calls are Linux-specific.

**SEE ALSO**

[inotify\\_add\\_watch\(2\)](#), [inotify\\_rm\\_watch\(2\)](#), [inotify\(7\)](#)

**COLOPHON**

This page is part of release 3.74 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 <http://www.kernel.org/doc/man-pages/>.