NAME

libinotify, **inotify_init**, **inotify_init1**, **inotify_add_watch**, **inotify_rm_watch**, **libinotify_set_param**, **inotify_event**, - monitor file system events

SYNOPSIS

#include <sys/inotify.h>

int
inotify_init();

int
inotify_init1(*int flags*);

int
inotify_add_watch(*int fd*, *const char *pathname*, *uint32_t mask*);

int
inotify_rm_watch(*int fd*, *int wd*);

int
libinotify_set_param(*int fd*, *int param*, *intptr_t value*);

DESCRIPTION

The **inotify_init**() and **inotify_init1**() functions create an inotify instance and returns a file descriptor referring to the instance. **inotify_init1**() function is similar to **inotify_init**() except that it takes additional flags parameter whose values can be:

IN_NONBLOCK Set I_NONBLOCK file status flag on the inotify file descriptor.

IN_CLOSEXEC Set FD_CLOEXEC flag on the new file descriptor. See O_CLOEXEC flag in open(2)

The function returns the file descritor to the inotify handle if successful otherwise return -1. Possible errorno values are -

EINVAL Invalid flag value passed.

EMFILE System wide limit of inotify instances reached.

ENFILE System limit on total number of fd’s reached.

ENOMEM Insufficient kernel memory.

inotify_add_watch() function adds news watch to the inotify instance. List of possible masks are described below. If the watch for given filename already exists, it it updated with the new mask value passed. The function returns an integer called watch descriptor if successful otherwise -1.

Possible values for errorno are -

EACCES Permission for read access is denied for given file.

EBADF Invalid file descriptor.

EFAULT Pathname points outside process’s allocated address space.

EINVAL Invalid event mask passed.

ENOENT A component of path that must exist does not exist.

ENOMEM Insufficient kernel memory available.

ENOSPC User limit on total number of inotify watches has crossed or kernel failed to allocate a needed resource.

inotify_rm_watch() function removes watch wd from the instance described by file descriptor fd. The function returns zero on sucess and -1 on error. Possible errorno values are -

EBADF Invalid file descriptor fd.

EINVAL Invalid watch descriptor wd.

libinotify_set_param() Libinotify specific. Replacement for Linux procfs interface. Set inotify parameter for the instance described by file descriptor fd. fd value of -1 is used for setting of global parameters. Possible param values are -

IN_SOCKBUFSIZE Size of communication socket buffer in bytes. Should match read(2) buffer size for libinotify event consumers. Lower values can cause partial event reads. Bigger values is just a wasting of memory. Default value is arbitrary, has been acquired from code sample in linux inotify(7) man page and seems to be very common among the inotify clients. Default value 4096 (exported as IN_DEF_SOCKBUFSIZE)

IN_MAX_QUEUED_EVENTS
Upper limit on the queue length per inotify handle. linux‘s /proc/sys/fs/inotify/max_queued_events counterpart. Default value 16384 (exported as IN_DEF_MAX_QUEUED_EVENTS)

IN_MAX_USER_INSTANCES
Global upper limit on the number of inotify instances that can be created. linux‘s /proc/sys/fs/inotify/max_user_instances counterpart. Default value 2147483646 (exported as IN_DEF_MAX_USER_INSTANCES)

inotify_event structure

```
struct inotify_event {
  int        wd;      /* Watch descriptor */
  uint32_t   mask;    /* Mask of events */
  uint32_t   cookie;  /* Unique integer associating related events */
  uint32_t   len;     /* Size of name field */
  char       name[];  /* Optional null-terminated name */
};
```

inotify events -

Following are the masks supported by libinotify implementation.

IN_OPEN	File was opened.
IN_ACCESS	File was accessed (read).
IN_ATTRIB	Metadata changed.
IN_CREATE	File/directory was created in watched directory.
IN_CLOSE_WRITE	File opened for writing was closed.
IN_CLOSE_NOWRITE	File not opened for writing was closed.
IN_DELETE	File/directory in watched directory was deleted.
IN_DELETE_SELF	Watched file/directory was deleted.
IN_MODIFY	File/Directory was modified.
IN_MOVE_SELF	Watched file/directory was moved.
IN_MOVED_FROM	A file in watched directory was moved out.
IN_MOVED_TO	A file was moved into watched directory.
IN_ALL_EVENTS	Bit mask of all the above events.
IN_MOVE	Equal to IN_MOVED_FROM|IN_MOVED_TO
IN_CLOSE	Equal to IN_CLOSE_WRITE|IN_CLOSE_NOWRITE

IN_DELETE_SELF and IN_MOVE_SELF can occur only for watched file/directory. Other events can be marked for a file/directory in a watched direcotry. In that case the name of the file for which event is generated can be read by ’name’ field in inotify_event structure.

Following are additional bits that can be set in mask when calling **inotify_add_watch() -**

IN_DONT_FOLLOW
Don’t derefernce path name if its symlink.

IN_EXCL_UNLINK Do not generate events for unlinked childrens. (Currently not supported).

IN_MASK_ADD Add event mask for watch for given pathname.

IN_ONESHOT Remove watch after retrieving one event.

IN_ONLYDIR Only watch the pathname if it is a directory.

Following bits may be set by mask field returned by read(3)

IN_IGNORED Watch for removed (explicitely, revoked or unmounted).

IN_ISDIR Subject of this event is a directory.

IN_Q_OVERFLOW Event queue has overflowed.

IN_UNMOUNT File system containing watched file/directory was unmounted.

SEE ALSO

read(3)

HISTORY

inotify first appeared in Linux 2.6.13